HP Solve
Calculating solutions powered by HP

Issue 28, July 2012
Welcome to the twenty-eighth edition of the HP Solve newsletter. Learn calculation concepts, get advice to help you succeed in the office or the classroom, and be the first to find out about new HP calculating solutions and special offers.

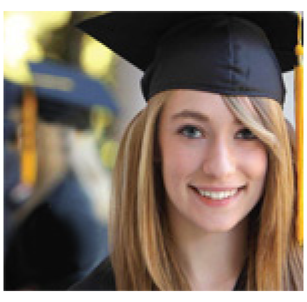
Download the PDF version of articles

Join our Facebook Fan Page

Contact the editor

In the spotlight

Graduate to HP Calculators
Don't get caught with a high school calculator in a college classroom. Get your new HP 10bII+ or 50g graphing calculator during back to school season.

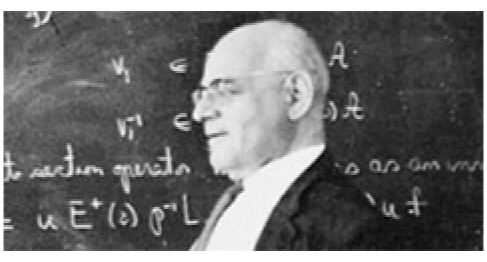


Read more

Your articles

The HP China Project
HP has partnered with the China Ministry of Education to introduce graphing calculators and data collection technologies into high school math classes, enhancing education.

Read more



Ostrowski's Method for Finding Roots
Namir Shammas

Namir explores an improved Newton's method for finding roots that all students learn in school. The Ostrowski method is examined, explained and exampled with programs for the HP 39gII and the repurposed HP-30b.

Read more

Store and Recall on HP Calculators
Richard J. Nelson

Here is a historical review of how HP has managed the use of data registers and memory usage on their first decade of calculators, the '70's. Many of these early methods are still in use today.

Learn more

Measuring Calculator Current
Richard J. Nelson

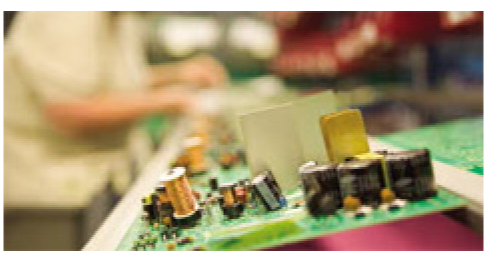
You press the "ON" key and nothing happens, now what? You replace the batteries and still have a blank display, do you throw it out and buy a new one? The most powerful technique to knowing what is or is not going on in your calculator is to monitor the current it draws.

Learn more

Is that (calculator) Answer Correct?
Richard J. Nelson

How do you know that the answer you have just calculated is correct? A few easy to learn (or review) techniques may be applied to approximate the answer so that you could even get by without the use of a calculator if necessary.

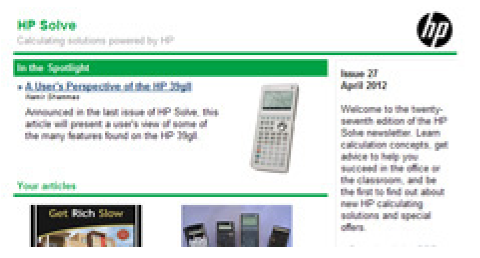
Read more



Calculators and Prefix Units
Richard J. Nelson

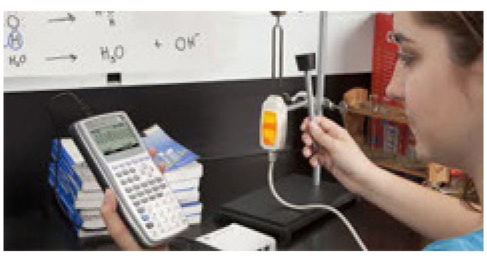
Calculators must be easy to use. Here is an idea for making them even easier and more convenient for problem solving. What do you think of this idea?

Read more



From the Editor
Learn more about current articles and feedback from the latest Solve newsletter including a new One Minute Marvels and HP user community news.

Learn more



Calculator Usage – Is Efficiency Important?
Richard J. Nelson

One of the reasons why calculators are still popular is their efficiency. This article reviews what this means and dispels some of the arguments made to discount calculator usage.

Read more

Graduate to HP Calculators

HP Solve #28 page 2

Article Next →

Return to top

Graduate to HP

HP 50g

HP 50g High-End Graphing Calculator was developed by engineers for engineers. Tackle complex problems in advanced math and engineering courses with an extensive equation library, multiple equation solver, built-in mathematical constants, a range of unit conversions, and 3D graphing capabilities.

HP 10bII+

HP 10bII+ All-Purpose Financial Calculator has every function you need to master business, finance and accounting courses plus a robust set of statistics, trigonometric, algebraic and math functions to carry you through general education math and science requirements.

For more information, visit www.hp.com/calculators

The HP China Project

HP Solve #28 page 4

THE HP CHINA PROJECT

Enhancing Education Through Strategic Partnerships

In 2008, HP partnered with the China Ministry of Education to introduce graphing calculators and data collection technologies into high school mathematics classes as an effort to enhance student comprehension of mathematic principles. Guided by the training and academic research of Professors Wang and Cao from Beijing Normal University, over 120 pilot schools in 15 Provinces throughout China are now using HP 39gs graphing calculators and the HP Mobile Calculating Lab (MCL).

Wang and Cao determined through their research that the key goals of mathematics education are to further develop Educators, improve course curriculum content and evolve Chinese teaching models to focus on multiple representations of mathematical expressions and open-ended problem-solving. The HP 39gs graphing calculator enhances mathematical understanding by allowing students to easily toggle between numerical, table and plot views of any mathematical expression. Educators have found that using these technologies allows students to more easily understand the connections between these representations and their underlying mathematical objects.

The expression, “shu xing jie he,” (combine number with graph) has been widely accepted as the golden principle for mathematics teaching since the 1960’s and demonstrates that Chinese Educators share similar ideas about mathematical representation. A more recent project conducted by the Abant Izzet Baysal University, Integration of Handheld Technology with the New Mathematics Curriculum, provides an opportunity to explore how Chinese Educators and students use graphing calculators and the impact this has on how mathematics is taught and learned.

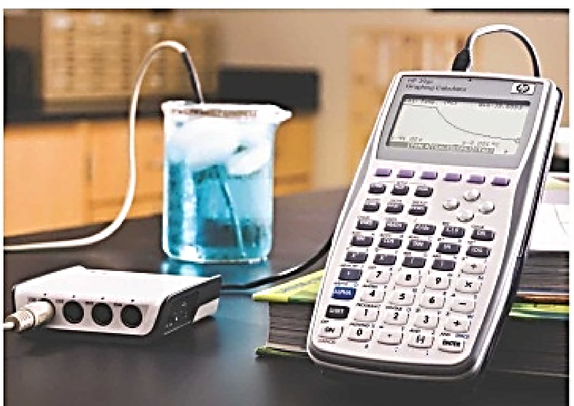
The HP—China MOE Project introduced an approach now known as "3Y's" from the Mandarin words for each concept that begin with the sound of 'Y'. The "3Y's" stands for:

- Use it – Implement an effective teaching strategy that motivates students to use the graphing calculator and develop a high-level of proficiency.
- Use it frequently – Apply multi-varied mathematical examples with the graphing calculator in every lesson to achieve greater learning results.
- Use it properly – Use graphing calculators to investigate and represent mathematical concepts to create enhanced learning ecosystems.

The HP—China MOE Project along with the "3Y's" model has raised teacher expectations of students; and students have met the challenge by responding with creativity and advanced reasoning. This new found approach to the teaching and learning of mathematics is quickly taking hold in communities throughout China.

To offer ongoing support, HP sends experts from an international community of math teachers to gather at summer and winter camps in China to share ideas, resources and pedagogy. One of the key highlights of these camps is the young scholar's math competition, which provides students greater opportunities to develop high-level insight to difficult concepts such as mathematical programming.

> "The HP China project has allowed for excellent research, the building of solid teaching pedagogy, student motivation and understanding of mathematics and it has provided many educators with a newfound method of teaching."
>
> *—Professor C P Wang (ATCM 16, Bolu, Turkey, 2011).*



Both professors agree that the MCL is changing the way students learn and comprehend complex math and science problems. The MCL has become an essential tool for mathematics education in China.

Earlier this year, HP introduced the newest member of its graphing calculator family—the 39gII. This new calculator maintains HP's iconic plot, numeric and symbolic keys to facilitate easy toggling between mathematical representations. Chinese characters and full Chinese menus have also been added to comply with the China MOE requirements.

Scientific:

- 10s+ — Coming soon
- 300s+ — Coming soon
- 35s

Graphing:

- 39gii — NEW!
- 40gs
- 50g

Visit www.emcl.org.cn or www.hp.com/calculators for more information

Ostrowski's Method for Finding Roots

HP Solve #28 page 7

Ostrowski's Method for Finding Roots

Namir Shammas

Introduction

Mathematicians, statisticians, engineers, and scientists frequently deal with problems that require the calculation of one or more roots of functions. When HP launched its first programmable calculator, the HP-65 in 1974, the accompanying Standard Pac included a root-seeking program. The HP-65 Math Pac 1 also included another root-finding program. Over the following years, HP released new programmable calculators; each included root-seeking programs in their standard and math pacs. In 1978, HP launched the HP-34C that contained the very first built-in root-finding Solver. The Solver was a new, powerful, and convenient tool for calculating the roots of single-variable nonlinear equations. The Solver implemented a clever version of the Secant method. This method along with Newton's method remained the two favorite root-seeking algorithms for many decades among users of programmable HP calculators. In this article I present the Ostrowski root-seeking method which I consider a gem of an algorithm that has not received much publicity until somewhat recently. I also present listings for Ostrowski's method to run on the WP34S calculator (using a repurposed HP 30b) and on the new HP 39gII. Before I discuss the algorithm and present the two listings, I would like to first shed some light on the person of Ostrowski.

Ostrowski: A Short Biography

Alexander Markowich Ostrowski (1893 to 1986) was a talented Russian mathematician who was gifted with a phenomenal memory. He was studying math at Marburg University in Germany, when World War I broke out. He was therefore interned as a hostile foreigner. During this period, he was able to obtain limited access to the university library. After the war he resumed his studies at the more prestigious University of Göttingen and was awarded a doctorate in mathematics in 1920. He graduated *summa cum laude* and worked in different universities. He eventually moved to Switzerland to teach at the University of Basel, before the outbreak of World War II. Ostrowski fared well living in that neutral country during the war. He taught in Basel until he retired in 1958. He remained very active in math until late in his life.

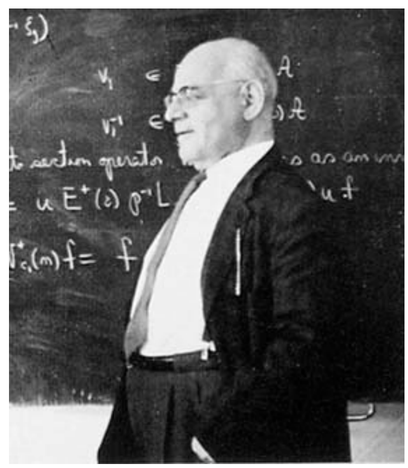


Fig. 1 – Alexander Ostrowski.

This gifted and prolific mathematician was engaged in various mathematical problems. The advent of computers catapulted Ostrowski to delve into numerical analysis. He studied iterative solutions of large linear systems. Ostrowski passed away in 1986 in Montagnola, Lugano, Switzerland. He had lived there with his wife during his retirement.

The Algorithm

Ostrowski tackled the problem of calculating a root for a single-variable non-linear function in a new way. Most of the methods we know perform a single refinement, for the guess of the root, in each iteration. Ostrowski's novel approach was to obtain an *interim refinement* for the root and then further enhance it by the end of each iteration. The two-step iterations in Ostrowski's method use the following two equations:

$$y_n = x_n - f(x_n) / f'(x_n) \tag{1}$$

$$x_{n+1} = y_n - f(y_n)(x_n - y_n) / (f(x_n) - 2f(y_n)) \quad \textbf{(2)}$$

Equation (1) applies the basic Newton algorithm as the first step to calculate y_n which Ostrowski uses as an interim refinement for the root. The iteration's additional refinement for the root comes by applying equation (2). Ostrowski's algorithm has a fourth-order convergence, compared to Newton's algorithm which has a second-order convergence. Thus Ostrowski's method converges to a root faster than Newton's method. There is the extra cost of evaluating an additional function call to calculate $f(y_n)$ in Ostrowski's method. This extra cost is well worth it, since, in general, the total number of function calls for Ostrowski's method is less than that for Newton's method.

Ostrowski's method has recently inspired quite a number of new algorithms that speed up convergence. These algorithms calculate two and even three interim root refinements in each iteration. While these methods succeed in reducing the number of iterations, they fail to consistently reduce the total number of function calls, compared to the basic Ostrowski's method. For example, Grau and Diaz-Barrero proposed the following equations for their algorithm:

$$y_n = x_n - f(x_n) / f'(x_n) \quad \textbf{(3)}$$

$$\mu = (x_n - y_n) / (f(x_n) - 2f(y_n)) \quad \textbf{(4)}$$

$$z_n = y_n - \mu\, f(y_n) \quad \textbf{(5)}$$

$$x_{n+1} = z_n - \mu\, f(z_n) \quad \textbf{(6)}$$

The algorithm starts with a Newton step using equation (3) to calculate y_n as the first interim refinement for the root. Equations (4) and (5) yield a second interim refinement, z_n. Equation (6) provides the iteration's final refinement for the root. The above equations show that each iteration requires two additional function calls needed to calculate $f(y_n)$ and $f(z_n)$. I compared the total number of function calls for Ostrowski's method with the Grau and Diaz-Barrero method for different test functions. The result is a mixed one, as neither method consistently performed better.

In the reference section you will find a reference to one of many books written by Ostrowski. The second cited reference is a paper by Walter Gautschi about the life, works, and students of Ostrowski. Gautschi himself is a mathematician and one of the last students of Ostrowski. Gautschi has published books in the field of numerical analysis. The third cited reference points to a recently published book written by him. I also included references to several articles, I found on the Internet, which present variants for the Ostrowski's method. I encourage you to search for these articles and learn more about these new algorithms.

The HP 39gII Listing

Table 1 contains the commented listing for the HP 39GII calculator. The table shows the following functions:

- The function **MYFX** which implements the code for calculating the mathematical function f(x).
- The function **OST** which performs the calculations for the Ostrowski method. This function makes several calls to function **MYFX**.
- The function **GO** which uses forms and message boxes to interact with the user. This function also calls function **OST** to obtain the number of iterations and the root value.

Table 1 – The HP 39gII Listing

Statement	Comment
EXPORT MYFX(X)	Define function for f(x)=0.
BEGIN	
X-10*LN(1+4*X2+2*X^4);	
END;	
EXPORT OST(X,T)	Define the function for the Ostrowski method. The parameter X is the initial guess for the root. The parameter T is the tolerance value.
BEGIN	
LOCAL DIFF,FX,X0,FY;	Declare multi-character local variables.
0→I;	Initialize the iteration counter.
REPEAT	Start the main iteration.
I+1→I;	Increment the iteration counter.
X→X0;	Assign a copy of X to X0.
MYFX(X)→FX;	Calculate and store f(X).
0.001*(ABS(X)+1)→H;	Calculate and store the increment h used to numerical evaluate the derivative.
H*FX/(MYFX(X+H)-FX)→DIFF;	Calculate and store the refinement for the guess.
X-DIFF→Y;	Calculate and store the value for y.
MYFX(Y)→FY;	Calculate and store the value for f(y).
Y-FY*(X-Y)/(FX-2*FY)→X;	Calculate and store the updated guess for the root.
UNTIL ABS(X-X0) <= T;	Did the solution converge?
{X,I};	Create a list containing the refined guess for the root and the iteration counter. This list is the result of the function OST.
END;	
EXPORT GO()	Function that drives the root-seeking program
BEGIN	
LOCAL LST;	Declare a local variable.
INPUT(T,"Tolerance","Tolerance = ","Enter tolerance for root",1E-8);	Prompt user with a form to enter the tolerance for the root. Store the tolerance value in variable T.
INPUT(X,"Initial Guess","Guess = ","Enter initial guess for the root", 50);	Prompt user with a form to enter the initial guess for the root. Store the initial guess in variable X.
OST(X,T)→LST;	Call function OST with the arguments X and T. Store the resulting list in list variable LST.
MSGBOX("Number of iterations = " + LST(2));	Display the number of iterations in a message box. Uses expression LST(2) to obtain the number of iterations.
MSGBOX("Root = " + LST(1));	Display the root in a message box. Uses expression LST(1) to obtain the value for the refined root.
LST(1);	Return the root.
END;	

The code for function **MYFX** is simple and requires a single statement for the tested function. The function **MYFX** has the parameter X to pass the value needed to calculate the mathematical function f(x).

The code for the function **OST** has the parameters X and T which pass the values for the initial guess for the root and the tolerance, respectively. Notice the following aspects about this function:

- The **LOCAL** statement that declares a number of local variables that have multi-character names.
- The **REPEAT-UNTIL** loop that implements the iterations needed to refine the guess for the root. The **UNTIL** clause tests the convergence criterion. The test determines if the absolute difference between the old guess for the root and the new one falls at or below the tolerance value.
- The function's return value which is the list **{X,I}**. This list contains the refined root value, X, and the number of iterations, I.

The code for the parameter-less function **GO** has the following three parts:

- Two **INPUT** statements that display input forms to obtain the values for the tolerance and the initial guess for the root.
- A call to function **OST**. The function **GO** stores the result of **OST** in the local list variable **LST**.
- Two **MSGBOX** statements to display message boxes showing the number of iterations and the refined guess for the root. The function uses the expression **LST(1)** and **LST(2)** to access the root value and number of iterations, respectively.

The function **GO** returns the refined root value. This approach allows you to store the root for additional inspection. You can invoke functions **MYFX**, **OST**, and **GO** from the command input line.

After you key in the three functions in Table 1, you can run the program by performing the following tasks:

1. Type **GO** at the command input line. The function **GO** first displays the **Tolerance** input form, shown in Figure 2. You can enter a new value for the tolerance and then press the **OK** menu option (which is associated with the F6 menu key), or simply accept the current tolerance value and press the **OK** menu option.

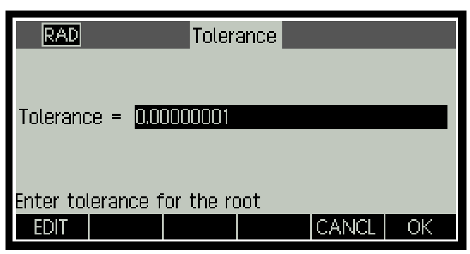


Fig. 2 – The prompt for the tolerance value.

2. The function displays the **Initial Guess** input form, shown in Figure 3. If the current guess is not 50, type in 50 and then press the **OK** menu option. Otherwise, just press the **OK** menu option.

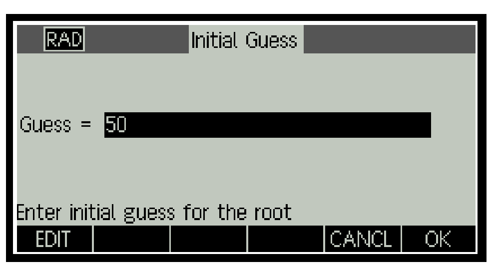


Fig. 3 – The prompt for the initial guess for the root.

3. The function displays the number of iterations, using the message box shown in Figure 4. Click the **OK** menu option to resume program execution,

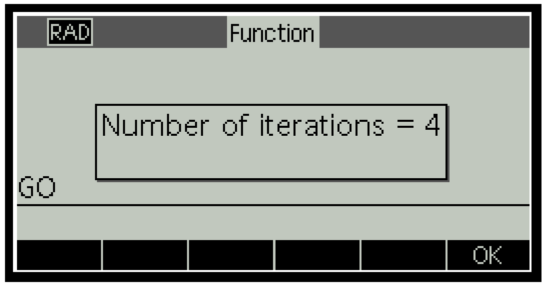


Fig. 4 – The number of iterations.

4. The function displays the value for the root, using the message box shown in Figure 5. Press the **OK** menu option to end the program.

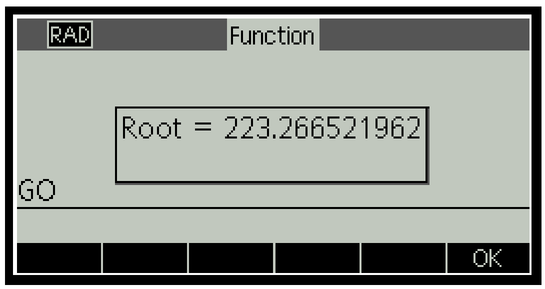


Fig. 5 – The refined root value.

I encourage you to edit function **MYFX** in Table 1 and replace the currently coded function with a different one. Table 2 shows a list of additional test functions. The table also includes columns that show the recommended initial root and the final root you get. Using the tolerance value of 1E-8 should be adequate for the test functions in Table 2.

Table 2 – A List of Additional Test Functions

Mathematical Function	*Initial Guess for the Root*	*Root*
$f(x) = x^3 + 4x^2 - 15$	3.0	1.6319808055661
$f(x) = x \exp(x^2) - \sin^2(x) + 3\cos(x) + 5$	-2.0	-1.2076478271309
$f(x) = \sin(x) - x/2$	1.1	1.8954942670340
$f(x) = 10 x \exp(-x^2) - 1$	1.0	1.6796306104284
$f(x) = \cos(x) - x$	10	0.73908513321516
$f(x) = \sin^2(x) - x^2 + 1$	0.1	1.4044916482153
$f(x) = \exp(-x) + \cos(x)$	0.1	1.7461395304080

The WP34S Listing

Let me present the listing, in Table 3, for my implementation of the Ostrowski method on a WP34S calculator. The program labels appear in red characters so they are easier to locate. The emulator I used for this article is running version 3 of the calculator software.

Table 3 – The WP34S Listing

Program Step	Comment	Program Step	Comment
LBL'OSTR'	Start and initialize the program.	x	
CLREG	Clear all the registers.	STO 02	Calculate small increment, h, used to numerically evaluate the slope.
EEX		RCL 00	
+/-		XEQ D	Calculate f(x).
8		STO 03	Store f(x).
STO 05	Store 1E-8 in register R05 as the initial tolerance value.	RCL 00	
RTN		RCL 02	
LBL D	Label used to code f(x)=0.	+	
LocR 003	Declare three local registers-R.00, R.01, and R.02.	XEQ D	Calculate f(x+h).
STO .00	R.00 = x	RCL 03	
x^2		-	
STO .01	R.01 = x^2	RCL 02	
x^2		/	
STO .02	R.02 = x^4	1/x	
RCL .00	Start evaluating f(x).	RCL 03	
1		x	Calculate the slope at x as (f(x+h)-f(x))/h.
RCL .01		STO- 01	Calculate y = x - f(x)/f ′(x).
4		RCL 01	
*		XEQ D	Calculate f(y).
+		STO 04	
RCL .02		RCL 00	
2		RCL 01	
*		-	
+		x=0?	
LN		GTO 01	
1		x	
0		RCL 03	
x		RCL 04	
-		STO+ X	
PopLR	Remove local registers.	-	
RTN		/	
LBL A	Label used to rerun the program.	RCL 01	
CLα	Clear the alpha register to start building new text.	x⇆Y	
α'TOL'		-	Calculate y-f(y)(x - y)/(f(x) - 2 f(y)).
α'ER?'		RCL 00	Recall older x.
RCL 05	Push current tolerance value in the stack.	x⇆Y	
PROMPT	Prompt for the tolerance value.	STO 00	Store new x.
STO 05		-	

Program Step	Comment	Program Step	Comment
LBL B	Label used to simply enter a new guess for the root.	ABS	
CLα	Clear the alpha register to start building new text.	RCL 05	
α'GUE'		x⇆Y	
α'SS?'		x>Y?	|new x – old x| > tolerance?
PROMPT	Prompt user to enter the guess for the root.	GTO 00	Resume iterations.
STO 00		LBL 01	
0		CLα	Clear the alpha register to start building new text.
STO 06	Initialize the loop counter.	α'ITE'	
LBL 00	Start of the main loop.	α'R='	
INC 06	Increment iteration counter.	RCL 06	
RCL 00		PROMPT	Display the number of iterations.
PSE 10	Display intermediate guess for the root.	CLα	Clear the alpha register to start building new text.
STO 01	Initialize y = x.	α'ROO'	
ABS		α'T='	
1		RCL 00	
+		PROMPT	Display the refined guess for the root.
EEX		GTO B	
3		END	
+/-			

If you are familiar with programming the HP-41C (and to a lesser extent other RPN programmable calculators like the HP-11C, HP-15C, and HP 35s) the above listing should be somewhat familiar to you. You will still find in Table 1 a number of new and different programming commands and features that are special to the WP34S. These new and different features are:

1. The program supports alphanumeric labels that are up to three characters long. The first command is **LBL 'OST'** which complies with this feature. This feature differs from the six character limit of alphanumeric labels in the HP-41C.
2. The WP34S calculator has four user defined keys labeled **A**, **B**, **C**, and **D**. They help you to easily execute code that you place after **LBL A**, **LBL B**, **LBL C**, and **LBL D**, respectively. The program in Table 1 uses the labels for keys **A**, **B** and **D**. By comparison, the HP-41C has more than four user defined keys.
3. The program uses **LBL D** to code the target mathematical function f(x). You can evaluate the function at any value of your choice by entering that value and pressing the key **D**. The code for calculating f(x) uses the local variables feature. The command **LocR *n*** dynamically allocates n local registers. The command also makes available 16 local flags, regardless of the value of n. The WP34S allows you to directly access the first 16 local registers, using the dot as a prefix followed by two digits. Thus to store a value in the first local register, you use **STO .00**. Likewise, to recall a value from the second local register, you use **RCL .01**. Beyond the 16^{th} register, you must use indirect addressing. The above code uses the command **LocR 3** to

dynamically allocate three local memory registers, R.00, R.01, and R.02. The command **PopLR** de-allocates the local registers. It is optional and I am including it for the sake of demonstration. By default, the local variables and flags in a subroutine are automatically removed when the subroutine reaches a **RTN** or **END** statement.

4. The program uses the alpha register to prompt the user for input and to display tagged/commented output values. By default, the WP34S simply appends characters to the alpha register. This feature is the reverse of HP41C's support for the alpha register, where inserted text automatically overwrites existing text in that register. Appending text in the HP-41C requires that you start with the special append character. Thus, to start inserting new text in the alpha register of the WP43S, you must first use the command **CLα** to explicitly clear the alpha register. Each program step can take up to three characters. So, for example to build the prompt text **TOLER?** I need two program steps. The first step inserts the first three characters **TOL**, while the second step inserts the last three characters **ER?**. The WP34S displays the character **α** to the left of the inserted text.

   You can insert one character per program step, but that is both wasteful and makes the listing harder to read. While entering letters in the alpha register is easy, entering the space, the question mark, the equal sign, and other punctuation characters requires some practice. The WP34S calculator stores various non-alpha characters in different menus and key combinations. I would like to point out that you can also append numbers to the alpha register. This feature, which I chose not to use in this article, can help to prompt program users in the input and display of array and matrix elements.

5. The program uses the **INC** command to increment the value of a memory (and also stack) register by one. The above listing uses the command **INC 06** to increment memory register 06 that stores the iteration counter. The WP34S has the **DEC** command to similarly decrement the value of a register by one. The **INC** and **DEC** commands are very handy in directly adding or subtracting 1 from a memory or a stack register without pushing 1 into the stack. This direct action does not disturb the stack. When all of the stack's contents are relevant, using **INC** or **DEC** is a welcome feature.

6. The **PSE *n*** command pauses the program for a specified number of tenths of a second. The above listing uses the command **PSE 10** to pause for one second.

The above listing has **LBL D** coded for the following function:

$f(x) = x - 10 \ln(1 + 4x^2 + 2x^4)$

Let's run the program to calculate the root with the initial guess of 50 and a tolerance value of 1E-8. The root is approximately 223.226. To run the program, perform the following steps:

1. Execute the command **XEQ 'OST'** to start and initialize the program. To rerun the program, simply press the key **A**.

2. The program prompts you for the tolerance value, as shown in Figure 6. The display shows the current tolerance value of 0.00000001. This is the initial value set by the program in step 1. If you enter a different value and rerun the program, you will see your most recent tolerance value. In the case of our example, accept the current tolerance value and press the **R/S** key.

Fig. 6 – The prompt for the tolerance value.

3. The program prompts you to enter the initial guess for the root, as shown in Figure 7. The value in the X register is a leftover from step 2. Just ignore it! Enter the value of 50 and then press the **R/S** key.

Fig.7– The prompt for the initial guess for the root.

4. The program pauses to display the series of improved guesses for the root. When it finds a refined root guess that is within the tolerance you specified, it displays the number of iterations, as shown in Figure 8. The figure shows how the alpha and X registers display a tagged (or commented) value. Simply press the **R/S** key to resume program execution.

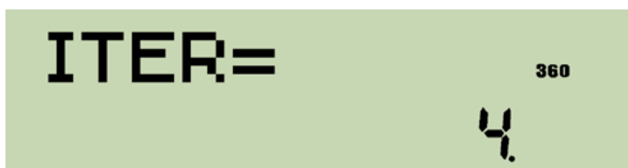


Fig. 8 – The number of iterations.

5. The program displays the improved guess for the root, as shown in Figure 9. Again, the figure shows the alpha and X registers displaying a tagged root value.

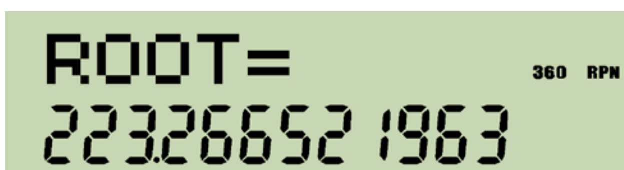


Fig. 9 – The refined root value.

You can press the **R/S** key and resume at step 3. Enter the initial guess of 1 to locate the other root for the function. The calculator displays the refined root of 0.025023481 in 4 iterations. Remember that you can evaluate the currently coded function f(x) by entering a value for x and then pressing the key **D**.

I encourage you to edit the code in **LBL D** in Table 3 to replace the currently coded function with the ones that appear in Table 2.

Observations and Conclusions

The article introduced you to Ostrowski's root-finding method. This fourth-order method excels over Newton's second-order method and Halley's third-order method. While Ostrowski's method has inspired new algorithms that further reduce the number of iterations, the basic Ostrowski method remains the most optimum as far as the total number of function calls. In addition to introducing you to the Ostrowski method, the article also presented two example listings: the BASIC-like program for the new HP 39gII and the RPN program for the repurposed HP 30b - WP34S. My hope is that the article will encourage you to use the efficient Ostrowski's method in future programs that you write.

References

1. A. M. Ostrowski, "Solution of Equations and Systems of Equations", Academic Press; second edition (1966).
2. Walter Gautschi, "Alexander M. Ostrowski (1893–1986): His life, work, and students", can be downloaded from http://www.cs.purdue.edu/homes/wxg/AMOengl.pdf.
3. Walter Gautschi, "Numerical Analysis", Birkhäuser Boston; 2nd ed. 2012 edition (December 6, 2011).
4. W. Kahan, "Personal Calculator Has Key to Solve Any Equation f(X)=0", Hewlett-Packard Journal, December 1979.
5. F. Soleymani and M. Sharifi, "A New Derivative-Free Quasi-Secant Algorithm For Solving Non-Linear Equations", World Academy of Science, Engineering and Technology 55 2009.
6. Xia Wang and Liping Liu, "New eighth-order iterative methods for solving nonlinear equations", Journal of Computational and Applied Mathematics 234 (2010) 1611-1620.
7. Reza Ezzati and Elham Azadegan, "A simple iterative method with fifth-order convergence by using Potra and Pták's method", Mathematical Sciences Vol. 3, No. 2 (2009) 191-200.
8. Guofeng Zhang, Yuxin Zhang, and Hengfei Ding, "New family of eighth-order methods for nonlinear equation", COMPEL volume 28, issue 6, 2009.
9. M. Heydari, S. M. Hosseini, and G. B. Loghmani, "ON TWO NEW FAMILIES OF ITERATIVE METHODS FOR SOLVING NONLINEAR EQUATIONS WITH OPTIMAL ORDER", Applicable Analysis and Discrete Mathematics 5 (2011), 93-109.

About the Author

Namir Shammas is a native of Baghdad, Iraq. He resides in Richmond, Virginia, USA. Namir graduated with a degree in Chemical Engineering from the University of Baghdad. He also received a master degree in Chemical Engineering from the University of Michigan, Ann Arbor. He worked for a few years in the field of water treatment before focusing for 17 years on writing programming books and articles. Later he worked in corporate technical documentation. He is a big fan of HP calculators and collects many vintage models. His hobbies also include traveling, music, movies (especially French movies), chemistry, cosmology, Jungian psychology, mythology, statistics, and math. As a former PPC and CHHU member, Namir enjoys attending the HHC conferences. Email him at: nshammas@aol.com

Store and Recall on HP Calculators

HP Solve #28 page 18

Store and Recall on HP Calculators – The First Decade

Richard J. Nelson

Introduction

Most calculators, even the lowest cost minimal featured ones, have the ability to store and recall at least one number. The first HP calculator, the scientific HP-35A[1], introduced in January 1972 had one storage register. Two keys, STO and RCL, were dedicated to this feature. See Fig. 1. In nearly every case the memory storage register has the same capacity as the display, and it is dedicated to the user's use.

Fig. 1 – HP-35A.

As memory technology advanced the number of data registers increased with later models offering hundreds of "data registers" available to the user. In some early HP models the use of data storage registers is shared by the calculator's functions. The user must be aware of the ones that may have unexpected data in them, and that they may even be cleared by the sharing, usually statistical, function(s).

This article will review the storing-and-recalling-of-data feature found on HP calculators. HP has been very creative in the way that data storage memory has been used.

Register sharing came first

The second HP calculator, the financial HP-80A[2], followed the HP-35A 13 months later. The scientific machine required the user to apply the formulas applicable to the broad fields of science and technology. The financial calculator, however, had the many finance formulas needed in the business world built in. This required 2-1/3 times the ROM memory. Memory was very important, but still technologically limited. Several of the HP-80A statistics functions even used some of the stack registers for their use. This made statistical function use a bit "tedious." The HP-80A single data register was called a "constant storage location" similar to the HP-35A. At the time, however, the HP-80A was many times faster and more accurate than any other financial computational method available to the average user.

Unique storage features happened quickly

The third HP calculator, the scientific HP-45A, followed the HP-80A three months later (5/1/73). The number of storage registers increased to nine and the STO/RCL operations increased in their capability.

These three machines are classical RPN calculators. The STO and RCL operations, however, were not RPN in the way that the machine used them. RPN is postscript logic wherein you provide the data followed by the operation. Storing a number, e.g. 7, in Register 1, should have the RPN sequence 7, 1, STO. Instead the sequence is 7, STO, 1.

This is explained by the inventor of HP calculators, Tom Osborne. Tom explained this during an interview of the HP-35A development team during a special event at HP Labs when the IEEE presented a plaque honoring HP on May 14, 2009. See ***HP Solve*** Issue 14, RPN Tip #14[3], near the end of the article. *"I probably would have used postfix on store if we had more than 10 storage cells (0-9), but "STO N" seemed much more easily understood than "N STO". However with more than 10 numeric memory cells then RPN would have won because it saves a keystroke. "STO 11" would have to be "STO 11 Enter" vs. "11 STO"."* The HP-45A was the first to add the feature of register arithmetic. Fig. 2 shows how the HP-45 Owner's Handbook explains register arithmetic.

Performing Register Arithmetic

Arithmetic operations (+, −, ×, ÷) can be performed between a data storage register and the **X**-register (display). To modify the contents of the storage register, press STO followed by the applicable operator key (+ , − , × , ÷), then the number key specifying the storage register. For example, store 6 in register R_1 then increment it by 2.

Press:	See displayed:	
6 STO 1	6.00	$6 \rightarrow R_1$
2 STO + 1	2.00	$r_1 + 2 \rightarrow R_1$

To see what is now stored in Register R_1,

Press:	See displayed:	
RCL 1	8.00	$r_1 \rightarrow$ **X** (display)

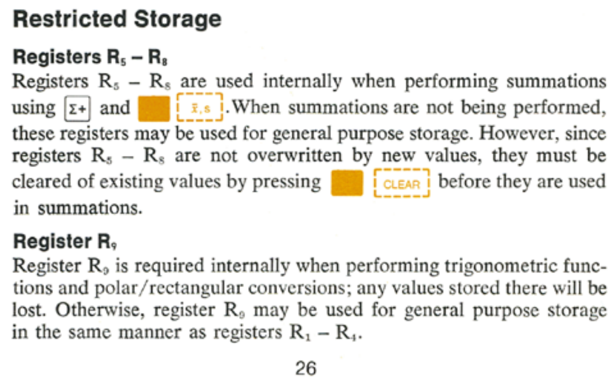
27

Fig. 2 – The HP-45 Owner's Handbook explains register arithmetic on page 27.

Tom's brilliant engineering set the tone very early for keystroke efficiency, a hallmark of HP calculators.

Register arithmetic makes perfect sense when memory is scarce (expensive). A data register is then able to serve multiple purposes. The most obvious example of multiple date register use is as a summing register. The HP-45A is not programmable, but using register arithmetic in a program could increase program efficiency as well.

The nine storage registers of the HP-45A (R1 – R9) were not completely without restrictions. R5 – R9 were shared by other calculator functions. See Fig. 3 text from the Owner's Handbook, OH, for the details.



Restricted Storage

Registers $R_5 - R_8$

Registers $R_5 - R_8$ are used internally when performing summations using Σ+ and $\bar{x}, s$. When summations are not being performed, these registers may be used for general purpose storage. However, since registers $R_5 - R_8$ are not overwritten by new values, they must be cleared of existing values by pressing CLEAR before they are used in summations.

Register R_9

Register R_9 is required internally when performing trigonometric functions and polar/rectangular conversions; any values stored there will be lost. Otherwise, register R_9 may be used for general purpose storage in the same manner as registers $R_1 - R_4$.

26

Fig 3 – HP-45A OH Restricted Storage description.

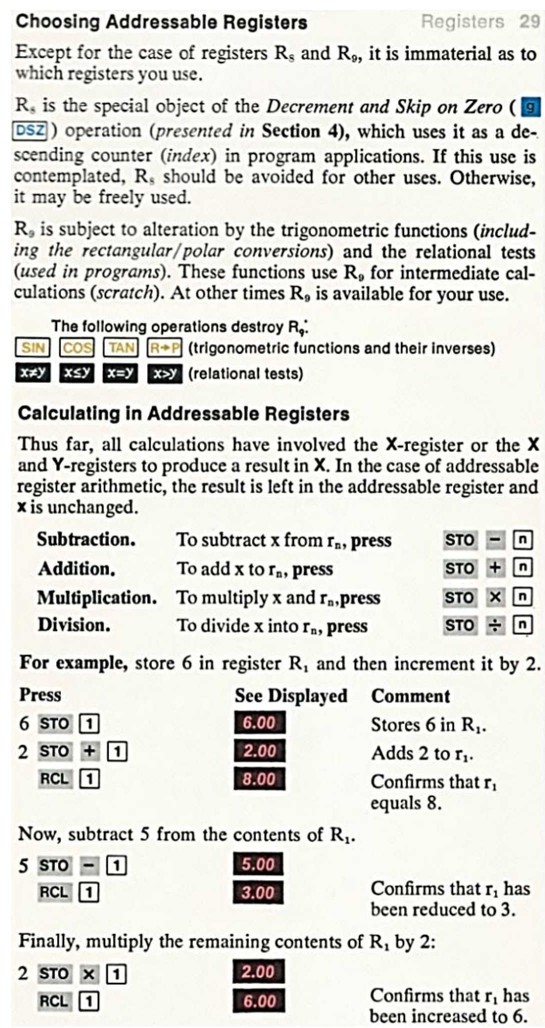
Recall arithmetic applied as well. The HP-45A Owner's Handbook explains. "*Conversely, to alter the X-Register (displayed value) without affecting the contents of the data storage register or the other stack registers, press RCL, the applicable operator, then the number key specifying the storage register.*"

The statistical registers may be also be used as summing registers (R7 ∑x & R8 ∑y). Using the ∑ key to store the sums of two numbers at the same time is very keystroke efficient without using register arithmetic.

The next major advancement in HP calculators was to add programming with the introduction of the HP-65A on January 1, 1974. This machine caused an explosion of HP calculator user activity(4). Memory was still relatively expensive and the HP-65A, like the HP-45A, had nine data registers (R1 - R9).

Fig. 4 at the right provides the HP-65A Owner's Handbook details of the usage of the nine data registers - on pages 29 & 30.

The HP-65A was programmable with enough memory for 100 program instructions, and it is possible to key data into program memory. Each digit, sign, decimal point, and EEX required a program instruction. Depending on the number of digits, etc. the 100 program instructions could consist of the data followed by a STO N. This data is then recorded on a magnetic card. To store data in the nine data registers you read the card and press the R/S key. This process is much easier and faster than storing the data as needed. An example would be English-Metric conversion factors for a program. Memory was very limited and the magnetic card off-line storage was a powerful asset for a machine that lost all memory when the power switch was turned off.



Choosing Addressable Registers Registers 29

Except for the case of registers R_8 and R_9, it is immaterial as to which registers you use.

R_8 is the special object of the *Decrement and Skip on Zero* (g DSZ) operation (*presented in* **Section 4**), which uses it as a descending counter (*index*) in program applications. If this use is contemplated, R_8 should be avoided for other uses. Otherwise, it may be freely used.

R_9 is subject to alteration by the trigonometric functions (*including the rectangular/polar conversions*) and the relational tests (*used in programs*). These functions use R_9 for intermediate calculations (*scratch*). At other times R_9 is available for your use.

The following operations destroy R_9:

SIN COS TAN R→P (trigonometric functions and their inverses)

x≠y x≤y x=y x>y (relational tests)

Calculating in Addressable Registers

Thus far, all calculations have involved the **X**-register or the **X** and **Y**-registers to produce a result in **X**. In the case of addressable register arithmetic, the result is left in the addressable register and **x** is unchanged.

Subtraction.	To subtract x from r_n, press	STO − n
Addition.	To add x to r_n, press	STO + n
Multiplication.	To multiply x and r_n, press	STO × n
Division.	To divide x into r_n, press	STO ÷ n

For example, store 6 in register R_1 and then increment it by 2.

Press	See Displayed	Comment
6 STO 1	6.00	Stores 6 in R_1.
2 STO + 1	2.00	Adds 2 to r_1.
RCL 1	8.00	Confirms that r_1 equals 8.

Now, subtract 5 from the contents of R_1.

5 STO − 1	5.00	
RCL 1	3.00	Confirms that r_1 has been reduced to 3.

Finally, multiply the remaining contents of R_1 by 2:

2 STO × 1	2.00	
RCL 1	6.00	Confirms that r_1 has been increased to 6.

Fig. 4 – HP-65 OM register arithmetic.

Register use is "re-defined"

The next (second) finance machine, the HP-70A(5), introduced 8/1/74, had two data registers called the M and K registers. Fig. 5 shows how the keys were labeled and Fig. 6 shows how they were described.

The M register is the familiar data register. You store by pressing N, STO, M. You recall by efficiently pressing M. The M+ key provides a summing function. The K register is similar except that K has a default value of 12, an obvious monthly number, stored in it when the machine is turned on. The M+ key idea is used on machines such as on the HP-01, HP-10A, HP-10B and HP-21.

Fig. 5 - HP-70A STO/RCL keys.

Storing and Recalling Numbers

Two general purpose memory registers are provided apart from the stack. One memory register (M) is primarily useful when you need to store data for use in subsequent problems or for accumulation. The other memory register (K) will probably be most convenient to use in repetitive calculations where one value remains the same (constant).

Although the store and recall functions work identically for both memory registers, since each has been assigned special purpose features to extend its use, we have described them separately.

Fig. 6 – HP-70A OH description of data registers.

Classical RPN data register addressing method is expanded further

The Classic RPN issue of addressing registers, e.g. R1 – R9 is explained by Tom Osborne in Note 3. Statistical functions need registers, conversion and scientific constants need registers. Users found that entering numeric data from a program was relatively slow so computational variables stored in a data register offered several advantages. These, and many other justifications, provided the need for additional numbers of data registers. The programmable scientific HP-55A calculator was introduced a year after the HP-65A (1/7/75).

The feature that made the HP-55A famous was its crystal controlled timer. HP-45A users discovered that the code for the timer was included in the HP-45A(6), but without a crystal or HP documented means to access the code. Many users installed their own quartz crystal to have their own HP-55A timer.

The HP-55A did not have register arithmetic, and program memory was limited to 50 lines (instructions). The number of data registers, however, more than doubled, (from 9 to 20) what was previously available. The registers were addressed as R0 – R9, and R**.**0 – R**.**9). The method of using zero and the decimal point retained the by-then-well-known method of addressing data registers. This provided the "lowest keystroke count" implementation that HP is so well known for.

Fig.7 - HP-55A.

Even more data registers

The next programmable scientific calculator, the HP-67A(7), added 6 additional storage registers and a new scheme that changed how they were used. Keystroke efficiency must have been strongly considered as shown in Fig. 9.

The top row of five keys were identified A – E. Associated with these keys were five primary data registers plus a 6th "I" register. See Fig. 8 below. A second 10 registers were called secondary registers, R0 – R9. If you wanted to store N into register A you put N in the X register and pressed STO, A. e.g. N, STO, A. The "I" register required a shift STO. If you wanted to store the number into a secondary register you pressed N, STO, 0. In this way 16 registers (A-E, I, & R0 – R9) could have values stored and recalled by pressing two keys.

F8 – HP67A Keys.

A second set of ten secondary registers, Rs0 – Rs9, are protected and shared with other calculator functions. A special function, P⇄S, exchanged the contents of the two sets of secondary registers. The STO and RCL keys would operate as they did normally. Fig. 9 below shows how all the data registers are designated.

The "I" register is a special indirect addressing register that is also shared between the user and the machine. A related STO & RCL calculator function is a register review feature introduced in the HP-67/97 calculators.

Another feature related to the SRO/RCL functions is the clear register function. This function clears all of the primary data registers. To clear the secondary registers the P⇄S key is pressed and the clear register function executed again.

Another advantage the HP-67A offered over the HP-65A is that the data registers were recorded onto a magnetic card.

From the above description it should be clear that keeping track of which register values are in the R0 – R9 primary vs. secondary registers is up to the user.

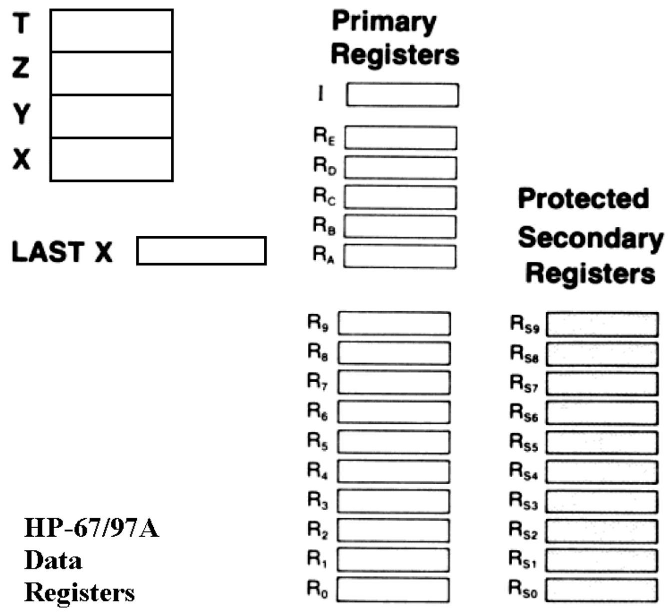


Fig. 9 – Data register designations of the HP-67/97A.

One register six keys

The STO and RCL features of the seven machines described above comprise the calculators that HP made in the first five years of calculator manufacturing. These are the machines of the historical Classic series.

Fig. 10 - Top: HP-21 STO and RCL keys.

Right: Shifted Memory function keys provide register arithmetic.

Following the Classic series is the Woodstock series of HP calculators. These were smaller and lighter and they picked up where the classic series left off. There are four scientific models and two business models. The Woodstock series is a transition series in that Continuous Memory first appeared towards the end of the series. The low priced HP-21 was the entry Scientific model having an M data register. The next scientific model was the HP-25 and it was programmable with 49 steps. It has eight data registers, R0 – R7, storage register arithmetic, and register sharing with the statistical registers (R3 – R7). The HP-25A was famous because it was cost effective, and the HP user community considered it a great challenge to squeeze many advanced level programs into its limited memory. Keying 49 steps for a program wasn't a tremendous time challenge and was reasonable because turning the calculator off cleared all memory.

Along with the HP-25A came the third finance calculator, the HP-22, with its ten data registers R0 – R9. Register arithmetic was also provided, and R5 – R9 were shared with the statistical functions.

The next calculator was a more capable (and more expensive) finance calculator - the HP-27. This was a special calculator because of its unusual mix of business and scientific functions. The HP-27 has 10 data registers, R0 – R9, with full register arithmetic. A display indicator "OF" indicates when register values exceed 9.9999999 x 10^{99}. A clear storage register function clears all registers. R4 – R9 are shared with the finance functions (R7 – R9), and the statistical functions ∑+ (R4 – R9). Even some of the stack registers are used/altered.

The penultimate member of the Woodstock series (1975 – 1977) was the HP-25C. The difference was the suffix letter "C" for Continuous Memory. This meant that data registers (and program memory) retained their contents when the calculator was turned off.

Continuous memory and even more data registers

The last machine of the Woodstock series is the HP-29C[8]. The HP-29C/19C has 98 program steps and 30 data registers designated as shown in Fig. 11.

Sixteen of the 30 are primary storage register addressed as R0 – R.5. The remaining 14 registers are indirect storage registers addressed by placing the register number (R0 – R29) in R0 and executing the STO I function. These 30 registers may have their values indirectly stored, recalled, and operated on with storage-register arithmetic. A number outside the range of 0 – 29 stored in R0 will error if the indirect function is performed.

The decimal part of the R0 register number is ignored and it may be negative for rapid reverse branching in a program. The HP29C also has a register clear function.

Automatic Memory Stack	Primary Storage Registers	Indirect Storage Registers
T	R_0	$R_{(16)}$
Z	R_1	$R_{(17)}$
Y	R_2	$R_{(18)}$
X	R_3	$R_{(19)}$
LAST X	R_4	$R_{(20)}$
	R_5	$R_{(21)}$
	R_6	$R_{(22)}$
	R_7	$R_{(23)}$
	R_8	$R_{(24)}$
	R_9	$R_{(25)}$
	$R_{.0}$	$R_{(26)}$
	$R_{.1}$	$R_{(27)}$
	$R_{.2}$	$R_{(28)}$
	$R_{.3}$	$R_{(29)}$
	$R_{.4}$	
	$R_{.5}$	

Fig. 11 – HP-29C Data Registers.

"Different" storage registers

The (first) HP-10[9], 7/1/77, is a one of a kind printing calculator obviously intended as an office calculator and is neither a business nor scientific calculator. It has two storage registers, The Accumulator and Memory. Per the Owner's Handbook: "*Basically, the accumulator is a memory that holds numbers while you perform other calculations*." The Accumulator is associated with the double high blue + /= key (see Fig. 12) and accumulates values as numbers are added or subtracted. This key is typical of adding type machines.

The HP-10A Memory is described as a "holding" bin or storage place for numbers. The complete keyboard is shown at the right in Fig. 12. Note the three "M" keys in the top row. Note that the gold shift key only has three functions associated with it.

Fig. 12 – Complete HP-10A Keyboard.

Less data memory for a lower cost

The last two years of the first decade (70's) of HP calculators brought us the Spice/Spike series of calculators. These machines are also called the "E" series because the first five models had an "E: suffix. Two of the three remaining models were unchanged and were the Continuous Memory versions with the same model numbers and a "C" suffix. The models were: Scientific; HP-31E. HP-32E, HP-33E, HP-33C, HP-34C. Business; HP-37E, HP-38E, HP-38C.

The "E" Suffix meaning was hinted at in the HP newsletter, HP Key Notes, as "Extensive low-end product line" Many users believe the "E" stands for Economy. Fig. 14 on the next page illustrates that the HP-31E was directed at students. $60 in 1978 is equal to $213.73 today[10].

The HP-31E Owner's Handbook was in two parts. The basic 17 page booklet and a 56 page "Solving Problems with Your Hewlett-Packard Calculator[11]." Data register usage is described in the later booklet which was also used with several "E" series machines. The HP-31E had four unshared data registers, R0 – R3. Fig. 13 shows the related data register keys. Note the CLEAR REG function.

The HP-32E was the next "E" machine and it was a step up from the HP-31E. The 32 was also nonprogrammable, but added more conversions, statistics and 15 data registers, R0 – R8 and R.0 – R.5. These registers also provided storage arithmetinc.

R.0 – R.5 and the Y-register were shared by the statistical registers.

The last “E” series scientific was the HP-33E. It has eight data registers, R0 – R7 with R2 – R7 shared with the statistical functions. Fig. 15 shows the HP-33E data register designations.

Fig. 13 – HP-31E STO and RCL related keys.

The First finance “E” series machines was the HP-37E. It has seven data registers, R0 – R6.

The previously four mentioned “E” machines were simple and non-programmable. The remaining four machines are more powerful and have more memory.

Fig. 14 - HP’s lowest cost 1978 scientific; the HP-31E.

Data memory shared with program memory

The HP-38E business calculator was HP’s first programmable business calculator introduced 5/1/78. With up to 99 lines of program memory it also shared statistical registers R1 – R6. The default memory allocation is eight lines of program memory and 20 storage registers. Each additional program line consumes part of a data register starting with R.9.

Each data register provide an additional seven lines of program. See Fig 16 and 17 for the HP-38E memory register allocations. When you add the ninth program line memory is automatically taken from the last data register, R.9. While this may seem strange based on today’s machines it is important to keep in mind that this was the state of the art in mid-1978. As shown in Fig. 15 the display was still power hungry LED’s and the batteries were rechargeable NiCad’s.

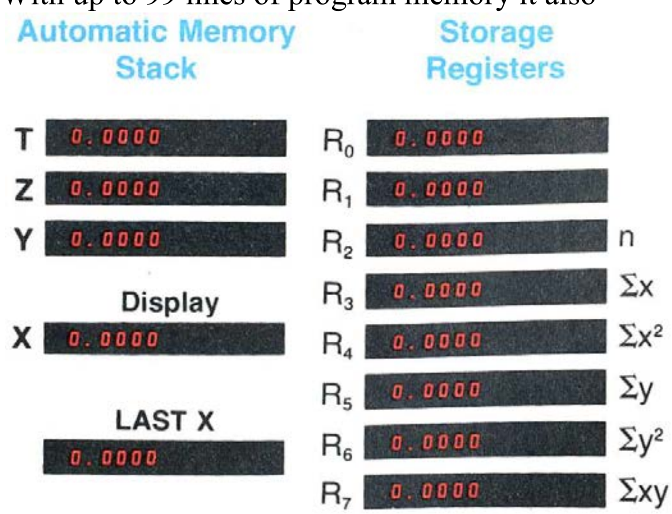


Fig. 15 – HP-33E register designations.

Continuous memory arrives at the end of the “E” series 7/1/79

The HP-33C is identical to the HP-33E except that program and memory do not disappear when the power is turned off. The HP-38C is identical to the HP-38E.

The big news of the “E” series was the scientific HP-34C. This was a new machine and it didn’t have an “E’ predecessor. Memory was a minimum of 70 program lines and a maximum of 210 program lines when the 20 data registers are similarly used as shown in Fig. 17. The major contribution for the HP-34C

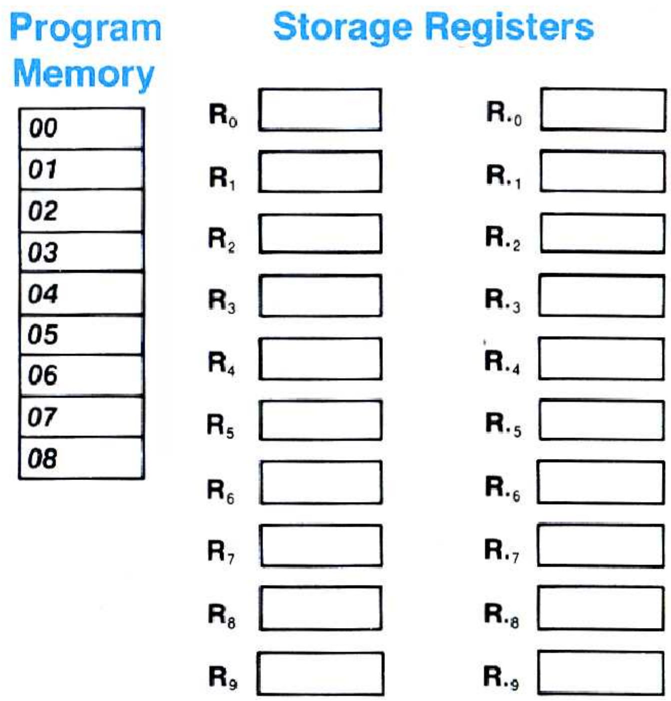


Fig 16 – HP-38E register allocation. 1 reg = 7 prog lines,

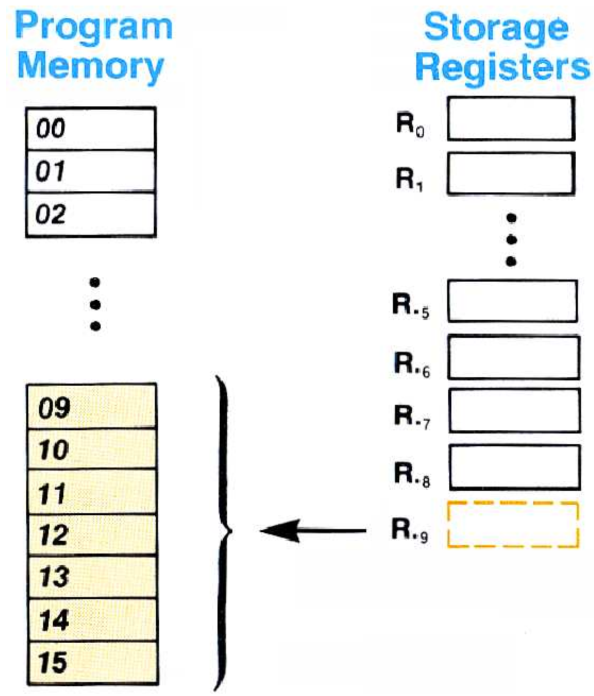


Fig. 17 - Additional prgm. lines consume data reg.

was the new and extremely powerful applications of Solve and Integrate.

Continuous Memory was a technology advancement that once started made it obvious that all HP calculators would soon be made using the CMOS process that needed such a low power that the machine is really never turned "off."

CMOS memory and HP-41 changes everything

1979 was a milestone year for HP calculators because of a new calculator concept represented by the HP-41C/CV/CX. The HP-41[12] utilized CMOS memory and it used an alphanumeric LCD display. It had a substantial amount of memory, and it had four I/O ports. The HP-41 is more than a calculator, it is a calculator system.

Prior to the HP-41 calculator models changed frequently, and a product life of about 18 months was normal. The HP 1979 calculator lab manager, Bernie Musch, suggested that because the HP-41 was a system it would have a product life of at least five years. He was happily mistaken when the HP-41 product life was double his expectations.

Because of the unique HP Corporate wide support given to the many accessories of the HP-41 it was used in every technical field from engineering to space[13] exploration and by everyone from students to doctors.

The first/basic model, the HP-41C, had 63 storage registers expandable (with memory modules or an HP-41CV) to 319 (R0 – R318) registers. These registers could be used for both program memory and data storage in a manner similar to what has been previously described except that the user specified the number of registers with a size function. Memory "reallocation" was not an automatic process. The memory organization is shown in Fig. 18. The user executes SIZE and then provides a number in response to a prompt similar to the STO function. Each data register uses seven bytes of program

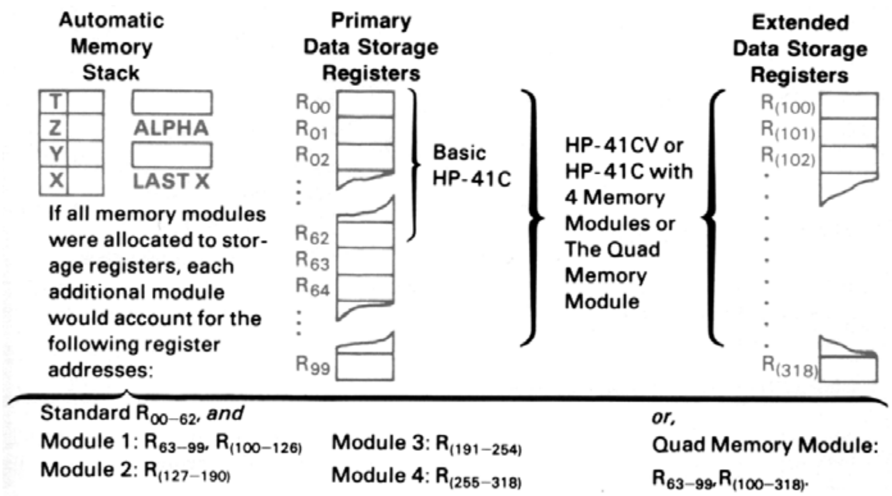


Fig. 18 – HP-41 Memory organization. The user sets the number of data registers with the SIZE function.

memory. Memory management is a bit involved because of the many combinations of models and memory modules for the four expansion I/O ports. The HP-41CX model provided additional modules built-in with maximum memory.

Data registers could be recorded on magnetic cards – the same ones as the HP-65A and HP-67A. Aside from the many 3rd party memory products there were other considerations for memory data registers e.g. the ALPHA register could also be used to store data if necessary. Another example is functions for the storing and recalling numbers into the stack registers. Another example is the use of synthetic programming[14] to access registers not normally available.

The HP-41 was a single line alphanumeric calculator and data storage requirements were impacted because a given register could store either a number or a small number (6) of alphanumeric characters. All registers (stack, LAST X, and Data) could be used for alpha characters.

The basic data registers are the primary storage registers R00 – R99. The STO and RCL operations require two digits. The display prompts for the register number with two underscore characters. The HP-41 has a register clear, CLRG, function. Storage register arithmetic may be applied to the primary data registers, R00 – R99. Storage register arithmetic may be performed on the stack as well. The HP-41 will also provide a register overflow message "OUT OF RANGE." A nice additional feature of the HP-41 is the ability to ignore error messages under the control (flag setting) of the user.

Data registers are used by the HP-41 for statistics functions. The normal six registers, however, are not fixed in that the user may set the starting register with the ∑REG function. The HP-41 keeps track of where these are and a CL∑ function will clear these six registers.

The HP-41 will use any register as an indirect addressing register with an IND function. When the sequence STO IND is executed the display will prompt with STO IND _ _. The user then provides the register number (or alpha designation) of the register to use for the register address. If you have SIZEd your machine to have more than 99 data registers you must address them indirectly i.e. three digit numbers R100 – R318. The alpha and stack registers may also be addressed indirectly. While not relevant to the data registers the HP-41 may take numerical arguments for certain functions indirectly, e.g. TONE, FIX, DSE, GTO including storage arithmetic and flags.

The HP-41 has two flags that may detect data entry, an ALPHA input flag (23) and a numeric input flag (22).

Summary, observations, and conclusions

Most calculators have the feature of providing one or more storage registers for storing numbers for use during a calculation or for future use (sums, constants, etc.). This article is a review of the use of data registers of the calculators HP made during the first decade of their calculators, the '70's.

Business and scientific calculator applications may require a surprising number of data registers and the calculators of this period ranged from one to 319 registers. In most cases these data registers are shared with the user by statistical functions that usually need six registers (and the stack) when they are used.

In addition to function sharing many HP calculators have a feature of performing storage, and in some cases, recall arithmetic in the data registers. In more advanced machines such as the HP-41 register arithmetic is allowed in the stack as well. Because the HP-41 is an alphanumeric machine alpha characters as well as data may be stored in data registers.

Programmable calculators often have a memory tradeoff feature wherein program memory may be used for data storage. In all cases HP has designed their data register usage to be as keystroke efficient as possible.

Store and Recall on HP Calculators – Notes

(1) *The classic model numbers; 35, 45, 65, 55, 67, 80, & 70 are suffixed with "A" to avoid confusion with later models, and to serve as a reminder of their history. The HP-35A designation was used internally by HP and is documented in* http://hhuc.us/2007/Remembering%20The%20HP35A.pdf

(2) *The HP-80A Owner's Handbook is a shirt pocket size similar to the HP-35A, but much thicker. It did not have an Index, but the detailed Contents provide a workable substitute.*

(3) *RPN Tip #14 may be downloaded at: http://www.hp.com/large/calculator/august09/the-rpn-stack-future-past-pt-2.pdf This is part 2 of an article discussing the Classical RPN stack. Most readers may not realize that the HP-35A predecessor, the HP 9100A, was actually a combination of infix and postfix. The electronics dictated the decision to go forward with Classical RPN in the machines that followed. Tom Osborne further explains.* "No one that I knew at HP Labs was familiar with RPN when I designed the 9100A. The green machine I took to HP was an interesting combination of infix for multiply and divide but post fix for add and subtract. The 9100A stack was high enough to solve most of the normal computations we encountered. With a bit of mental parsing on the input a 2 deep stack can solve any two operand problem, so we were more than covered. A really deep stack is required if one goes formal and leaves all of the operands in their original order and then relocates the postfix operators (as a full blown parser does)." *A full blown parser was implemented by others in much later HP machines.*

(4) *The worldwide HP Calculator user Group known as PPC (new models caused the name change) was founded in June of 1974. The club publications – long before the Internet – provided programs and technical information to greatly increase the usage of HP's calculators. The leadership of PPC continues today with HP calculator user activities such as small group meetings and the HHC Conferences.*

(5) *The HP-70A was an unusual business calculator for several reasons other than the data register usage. The HP-80A replaced the double wide ENTER key with SAVE. The HP-70A brought it back with a bright orange color. The double wide ENTER key was then retained on business calculators going forward. Collectors actively seek this machine because it is very difficult to find.*

(6) *The HP-45A timer could be invoked by pressing RCL and then pressing CHS 7 8 all at the same time. Once in timer mode, CHS toggled it between timings and stopped.*

Store and Recall on HP Calculators – Notes Continued

(7) The HP-67A was part of a "matched" pair of calculators. The other machine was the HP-97A which was a thin desk top machine with thermo printer. Programs (without printer functions) could run on either machine.

(8) The HP-29C was part of a "matched" pair of calculators. The other machine was the HP-19C which was a small desk top machine with thermo printer. The HP-29C was the first machine of the last RPN Stack list logic change as described in Table 1 on page 42 of **HP Solve** *issue 27.*

(9) HP has used the "10" model number more than any other number. Examples are: HP-10A, HP 10B, HP 10 BII, HP 10bII, HP 10BII+, HP-10C, HP 10s, and HP 10 Quick Calc. The space in Quick Calc is shown on the machine and was not actually intended as represented in the "named" models that followed. The HP calculator collector would have more than these eight models because of color variations sold with the same model number. For detailed information on the sharing of model numbers see **HP Solve** *issue 20 page 25. The "A" suffix is added here to indicate it is the first HP 10.*

http://h20331.www2.hp.com/hpsub/downloads/Newsletters_HP_Calculator_eNL_08_August_2010.pdf

(10) The cost in 2012 is based on the Federal Reserve Bank of Minneapolis consumer price index, cpi, calculator at: http://www.minneapolisfed.org/

(11) Every HHC 2010 attendee received a copy of this booklet with their conference proceedings because it a rare tutorial on Classical RPN. The manual printing cost was kept low by having the majority of the pages of the "E" series manuals in the 56 page "Solving Problems with Your Hewlett-Packard Calculator" which could be printed in larger quantities. The specific details of each model were in the much fewer pages individual model manuals.

(12) See **HP Solve** *Issue 16, "The HP-41 system – 30 years old" for a detailed description of this most popular HP calculator that continues to be "improved" even as this is being written. The HP-41 provided so many firsts (that are explained in the article) because it was a revolutionary leap in technology.*

(13) One of the best sites for HP calculator usage (all models) in space is at: http://hpinspace.wordpress.com/

(14) Synthetic Programming, SP, is the use of HP-41 synthetically assembled instructions that cause the machine to perform "illegal" operations. The PPC ROM User's Manual is the best place to start if you want to become familiar with the subject because SP is explained along with programs that will assist in the generation of the many synthetic instructions that are possible. An example of this power is machine-perfect Morse code. I wrote an HP-41 program that was published in the PPC Calculator Journal, February 1980 issue on page 50. Even using synesthetic tones the speed was limited to 6 words per minute. A more sophisticated SP technique of addressing memory produces a speed of 16 words per minute. A discussion of the technique to do this may be found in Keith Jarett's © 1982 book HP-41 Synthetic Programming Made Easy, page 151. The program in bar code is on page 183.

Measuring Calculator Current

HP Solve #28 page 29

Measuring Calculator Current

Richard J. Nelson

Introduction

The most important measurement of a semiconductor or semiconductor device is its current usage. I once visited an assembly house in Asia that packaged integrated circuits, IC's, (commonly called chips) into plastic packages. The finished IC's were then 100% tested using a massive computer controlled very expensive digital signal tester. I noticed that the tester wasn't in use and I asked why. The test engineer explained that they had found that they could simply measure the quiescent current and they could determine 99.9% of the time that the circuit was working. Besides, the hundreds (and often thousands) of specific digital conditions of the IC functions took much longer by a factor of at least ten.

Calculators are semiconductor products and measuring the current that they draw under various usage/mode conditions provides a good indication of what is happening. Pressing "on" with a blank calculator display just doesn't tell you very much.

Current is the most problematic basic electronics measurement – vs. voltage or resistance. Modern digital nultimeters, DMMs, are cheap, accurate, and readily available. All too often, however, you will find that the current range doesn't work – usually because of a blown internal fuse. Most electrical measurements are made by placing the probes across two points on the component or circuit under test. Measuring current, however, requires that you insert the meter into the circuit. This means that you must cut or un solder the circuit and this is usually not convenient.

Current Range

An interesting aspect of electronics is the very wide range of values that are involved. Current is measured in amperes. Your car starter may require 200 to 500 amperes to turn over the engine. A digital watch may require a millionth of an ampere to run. 1/1,000th of an ampere is a milliampere, mA, and 1/1,000,000th of an ampere is a microampere, μA. Nanoamperes and picoamperes are becoming more common specifications of IC currents in todays advanced low power IC's.

The current that a calculator will draw will vary greatly from low microamperes to low milliamperes. This range of 1,000 to 1 may instantly change with the press of a key. These highly variable electrical conditions add to the challenge of calculator current measurements. These considerations are:

(a) Inserting the meter into the circuit – a mechanical and technique issue.
(b) Using the correct meter and or range – an experience issue
(c) Keeping meter resistance under control – an expertise/understanding issue.
(d) Measuring and range changing – an experience/expertise/understanding issue.

All of these issues apply and the last one needs further explanation. When you turn on the switch of an electrical device current flows from the power source. Suppose there is something very wrong and the device is shorted or presents near zero ohms to the power supply. What prevents 1,000 amperes from flowing and turning all conductors cherry red? There are several reasons. The wiring involved may be very small – resistance of the wire may limit the amount of current that will flow. The power source may have a “built in” resistance that may limit the amount of current that will flow. Calculators are usually battery powered and the internal resistance of the physically small battery will usually limit the current. This source resistance is what the calculator electronics sees.

When you insert your current meter into the circuit between the battery and the calculator the calculator will see the meter resistance and the battery resistance in series. Both the meter and the battery will have its source resistance dependent on the conditions at the moment.

The battery source resistance will be quite low when it is first used[1] . The internal resistance will slowly increase as the battery supplies energy, and it will be quite high at the end of life.

The meter resistance will change when the range is changed. A high current range will have a much lower resistance than the low current range.

<u>Digital Circuits are “Time” Sensitive</u>

We have all heard that digital circuits are very fast. Most human activities such as key press times, reaction times, or neural response times are measured in the low millisecond range. Digital circuits respond a million times faster and the circuits that interface to the human world must be slowed down or time conditioned in order to avoid confusing the digital circuits.

Calculators have special turn-on circuits (reset circuits) that address the mechanical switch bounce of keys, switches, and users messing with electrical connections in trying to measure calculator current. All users have observed “strange” behavior when using their calculators, and they intuitively know that this is the nature of digital circuits. Computers are the same way and when in doubt – restart/reboot first.

<u>Breaking the Circuit</u>

The mechanics of breaking the battery circuit of a calculator means that you must get your two meter leads between the battery and the power input of the calculator. This is a mechanical problem and you just want to make a quick measurement. I am sorry, but making that “quick” calculator current measurement is not going to be very quick, or all too often, not very easy.

There is a resurgence in interest in the HP-41 because of Monte’s Newt project, the HP-41CL, so I will use the HP-41 as an example of applying the concepts discussed above. The first task is getting a DMM that has a low enough current range to measure at least as low as 1 µA up to 10 mA as a minimum range(s).

Fig. 1 – HP-41 N cell battery pack.

The HP-41 uses a spring loaded battery pack that holds four N cells connected in series to apply 6 volts DC to the calculator. See Fig. 1. Each calculator has its own specific mechanical requirements, but the basics described here will apply to most calculator current measurement situations.

I favor inserting a piece of double sided circuit board between the battery contact and the calculator. Newer machines will require a much thinner circuit board. The space is often very tight (especially with coin cells) and I have even used folded aluminum foil covered with scotch tape.
Fig. 2 shows the details of a piece of circuit board. The soldered leads, covered with heat shrink tubing, are shown in Fig. 3. This provides a physically strong assembly.

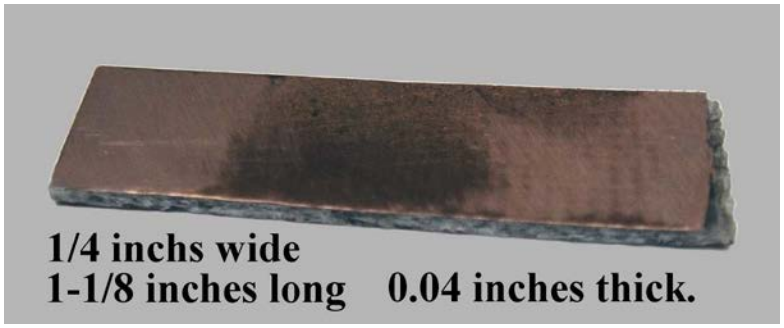


Fig. 2 – Double sided printed circuit board, PCB.

Fig. 3 – PCB with leads attached.

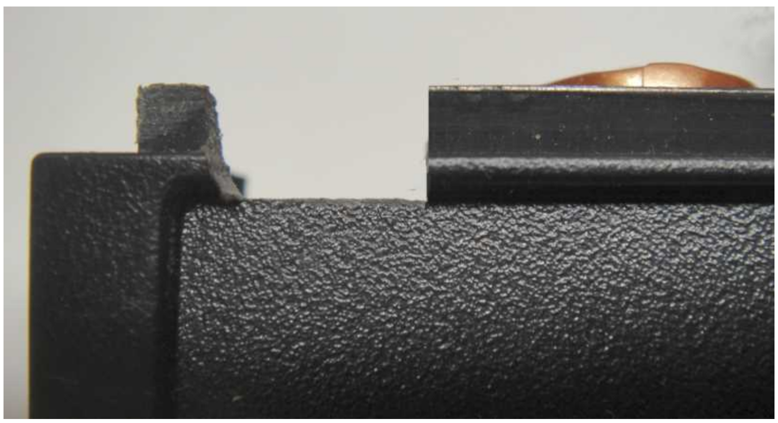


Fig. 4 – HP-41 Battery holder with filed notch – at negative end.

I have several HP-41's with spare parts. I modified one battery holder by filing a notch as shown in Fig. 4 by using a square file. The notch is on the negative end of the N cell holder. I did this many years ago and I don't remember why I chose this end.

Fig. 5 shows the battery pack installed with the "current probe" in place. The "current probe" may easily be pulled out. Inserting the probe requires holding it in place as the battery pack is slid in.

The internal "battery change capacitor" used for the HP-41 is large enough that it will retain a program in memory for many hours – often overnight. Because of this you may have to "reset" the machine by placing a jumper across the two outside battery compartment bumps and holding it there for at least 15 seconds. A calculator latch up may occur when you change current ranges on simple DMM's.

Here are some very simple measurements of an HP-41C to illustrate.

OFF[1] – 0.5 µA.
ON[2] - 0.53 mA.
Loop[3] – 4.19/4.22 mA
Pressing Key[4] – 4.13 mA.
Loop[5] – 4.26 mA

Notes:

[1] The value slowly drifts down to 0.1 µA which is the lowest resolution of the $3 (Harbor Freight sale) DMM. Comparing this DMM to a laboratory standard shows that it is quite accurate.

[2] In Fix 4 display.

[3] Using a simple (LBL 01, BEEP, GTO 01) loop.

[4] Any pressed key draws about the same value.

[5] Using a digit entry loop: LBL 03, 123456789 GTO 03.

If you duplicate the measurements using the same meter you will notice that the calculator won't turn on if the selected range is left on the lowest range (200 µA.). Use two switch positions higher in the 20 mA. position of the rotary switch.

If you want to avoid the meter switch range changing issue you may measure the current by measuring the voltage drop across a resistor inserted between the battery and the calculator. This is easy with the probe shown in Fig. 3. You just solder a one ohm resistor across the probe and you may change the meter range switch as much as you wish.

Fig. 5 – "Current probe" shown in place.

The low cost CEN-TECH DMM, however only has a 200 mV range as its lowest range and you will lose resolution. The one ohm is convenient because what you see in the display voltage wise may be converted to amperes. 1 mV is 1 mA. Even meters 10 or 20 times more expensive do not offer sub DC mV ranges. You may increase the resistor value to 10 ohms to add one more digit of resolution, but you are then back into the region where certain calculators may not turn on or operate correctly under all conditions. Certainly measuring microamperes is out of the question because a microvolt meter is much more expensive.

There is another consideration. If you are measuring sub millivolts you will have lead length (stray induced noise) issues and your electronics understanding will be further challenged.

What is a user to do? The technical issues are not too difficult and cost is not the issue. As I mentioned above measuring current is a challenge, but it does not have to be difficult.

Once again there is a mechanical solution that requires a bit of work. What you need to add to your probe set up is a normally closed push button switch connected across the two meter leads. When you want to measure the current you push the switch. Now you may futz with your meter and calculator as much as you wish without concern of switching transients disturbing the calculator and you just press the button to make your measurement. The calculator will always turn on when you need it to, and you may change ranges as needed (button is not pressed).

A More Modern Example

Many current HP calculators use coin cells for the power supply. Fig. 6 shows an example. Machines like the HP 20b, HP 30b, HP 10bII, and HP-15C LE use the CR 2032 coin cell. Fig. 6 and fig. 7 shows how the power connections are made. There is no physical space to use a "current probe" approach described with the HP-41 example.

For this situation the cells are removed and power is supplied using clip leads and miniature spring loaded connectors that are small enough to grab on and hold the small terminals of the cell holder. An external power supply of external coin cell holder is used for the power. The current meter (DMM) is connected in series with the power source.

Fig. 6 – Coin cell holder with electrical connections "buried."

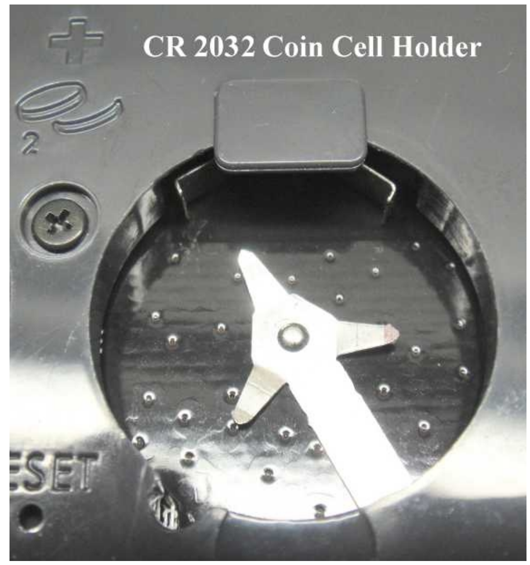


Fig. 7 – Small cell metal connections.

Fig. 8 – Miniature clips attached to the cell terminals; red positive, black negative.

Fig. 9 – Hook close up[2].

The primary current measuring procedure is as it always has been. Start at the highest current range and work your way down as needed. The issue with measuring calculator current is that the current will change with the various operating modes and conditions the calculator is used. The current drawn by the machine will range from sub microamperes to tens of milliamperes.

When you try to make a current measurement with your meter and it doesn't work you will soon learn that you violated this most important rule when making current measurements and you blew a fuse. Don't feel bad. EVERYONE has done it and EVERYONE will do it again. That is why making current measurements is more difficult than other fundamental electrical measurements.

How much current does your HP calculator draw? That is the subject of another article.

Measuring Calculator Current – Notes

(1) *Drawing that first, even very small (test), amount of current from a new battery will start the chemistry working and the battery will have its life determined from that moment forward.*

(2) *The hook shown is a lower cost knock off of the E-Z Hook model XG31. See the technical details at:* *http://catalog.e-z-hook.com/item/test-hooks/ll-categories-test-hooks-e-z-micro-hook-lock-hooks/xg31?*
These cost about $4 each and they are a vital requirement for making measurements on calculators.

Is That (calculator) Answer Correct?

HP Solve #28 page 36

Is That (calculator) Answer Correct?

Richard J. Nelson

Introduction

Before calculators were the accepted means for making numerical computations[(1)] the primary tool of choice for students was the slide rule. While the slide rule was capable of getting a correct answer it had a major flaw. The user had to keep track of the decimal point. Essentially this required that the problem solver needed to solve the problem twice. First you used a technique of approximation that would essentially provide the decimal point (order of magnitude) followed by the "numbers" of the answer provided by the slide rule.

If you talk to any teacher regarding the answers that students turn in for homework assignments or exams you will hear many stores about the wild answers that students turn in. They think that they have the right solution and with all those wonderful digits the calculator provides it must mean that the answer is correct.

An experienced person, such as the teacher, will have a practical sense of the correct answer. The student, however, doesn't have the benefit of an additional ten, or more, years of experience involving the application of the problem. Does the average ant weigh 0.11 oz.? Oh, that really was 0.00011 oz.- or 110 micro ounces (0.003 milligrams)..

Learning Math Computational Basics

In the early days of calculators there was a big discussion on weather calculators should be allowed in the classroom, and at what year. I remember attending a math conference session on this topic in a Los Angeles school. The presenter was making the argument for allowing calculators in the classroom when he just stopped talking. After a moment he asked "What time is it?" Like zombies everyone in the room looked at their wrists. He pointed out that we all have a natural sense of time, but since we have a mechanical device to provide an accurate time we use it. Why shouldn't we use a mechanical/electronic device to make accurate calculations? Looking back, the pro-calculator teachers "won[(2)]."

While calculators are able to take the drudgery out of mathematical calculations we still have to understand what the calculations mean. We still must understand the idea of division and multiplication. We MUST learn our multiplication tables (12 x 12) or as they used to do in the UK 13 x 13. We must learn, i.e. memorize, certain mathematical relationships. Can you quickly answer the following questions from memory?

1. What is 7 x 8?
2. What is 12 x 12?
3. What is 1/5 expressed as a decimal?
4. Which is larger 1/5 or 1/8 and by how much?

The reason for learning and memorizing certain mathematical relationships is very simple. Human beings learn best by going from the specific to the general. The specific is our reference point. We just know it and we then compare the unknown to our known (a standard if you wish) and we are better able to grow in our learning and understanding. I remember hearing this maxim. Yesterday I read a book and I leaned nothing, today I read a book and I learned something.

In addition to knowing the multiplication tables students should also learn the reciprocals of the single digit numbers including the "obvious" 10. One of the important reasons for memorizing reciprocals is in expressing percentages.

Digit, N	1/N	Digit, N	1/N	Digit, N	1/N
2	0.500	5	0.200	8	0.125
3	0.333	6	0.167	9	0.111
4	0.250	7	0.143	10	0.100

Mathematics is taught differently around the world. Mathematics is taught differently today than it was taught when I was in grade school. No matter how you leaned basic mathematics, e.g. the grid method or chunking, you have to know the basics mentioned above – multiplication tables and reciprocals.

Approximating the Answer Example

Part of the argument against the use of calculators in the classroom is that students will "loose" (or never learn) a practical sense of numerical calculations. There was a time that students were taught how to extract a square root by hand. Why is all this important? I am reminded of the primary maxim of the reporter. Never ask a question in an interview that you don't know the answer to. Making numerical calculations should follow the same (slightly modified) maxim, never use a calculator calculation that you don't know the approximate answer to.

I was reminded of an approximation technique I learned while eating lunch and thinking about an electronics problem. I needed to solve the equation that calculated the capacitive reactance of a capacitor using it as a 60 Hz. voltage divider.

$$X_C = \frac{1}{2\pi f C}$$

I had a 0.01 μF capacitor. What would be its reactance at 60 Hz? One μF is a millionth of a Farad. I then remembered that I really didn't need my trusty HP-15C, HP-48GX, or HP 35s. Expressing the value of C in scientific notation is 1×10^{-8}. Let's use a notation of 1E-8. The denominator $2\pi f$ is 2 x 3.14 x 60 or 6.2 x 60 or about 360 times the capacitance ≈ 360 x E-8. This is 3.6E-6. The reciprocal of 4 is 0.25 so the approximate answer is 0.25E6 or 25E4 or in normal electronics values 250 K ohms. When I make the calculator calculation and get 265,258.23849 ohms I am confident that all those digits are correct.

Reviewing Exponents

1. Approximate notation examples 1,000 = 1000E0 = 100E1 = 10E2 = 1E3. Notice how making the value before the E smaller by ten the value following the E gets larger by 10. Here is how small numbers work. 0.0001 = 0.0001E0 = 0.001E-1 = 0.01E-2 = 0.1E-3 = 1E-4. The idea is to reduce any number to a single digit with perhaps a single decimal digit. 2012 ≈ 2E3. Rounding is necessary. π ≈ 3E0. 265,258.23849 ≈ 2.7E5 or 3E5.
2. Multiplying two numbers expressed in scientific notation with exponents adds the exponents. Multiply 263 x 343. 3E2 x 3E2 = 9E4 ≈ 90,000. The calculator gives 90,209
3. Dividing two numbers expressed in scientific notation with exponents subtracts the exponents. Divide 4723 by 13.46 ≈ 5E3 ÷ 1E1 ≈ 5E2 or 500. (E3 – E1 = E2). The calculator gives

350.8915300. The approximate answer may seem a bit low and here is where experience and judgment may be applied in terms of how you round the numbers. The denominator is much smaller than the rounded value and the estimated result is expected to be larger.

Making a Guess Leads to the Exact

There are many methods and techniques for making numerical calculations and professionals who work with calculations will learn a few of them such as the rule of nines[(3)]. Books have been written on the use of the rule of nines in (accounting) validating numerical calculations.

Suppose you needed to know the average of the following numbers – these were measured values.

#	Value	
1	106	-4
2	108	-2
3	109	~~-1~~
4	110	Guess
5	111	~~+1~~
6	110	0

You first make a guess by examining the numbers. In this case the guess is 110. Next you note the differences between your guess and the values. The third step is strike out the ± values that cancel. This leaves a total of -6. There are six values so the average error of the guess is -1. 110 -1 is 109. The average value is 109 exactly.

Conclusion

Experience in problem solving provides a sense of what the answer should be. The student doesn't have that experience and he or she should develop techniques or methods to provide an approximation of the answer as a check for the answer obtained by the calculator. This article provides a few techniques for approximating the answer to making numeric calculations. Calculations involving exponentials or trigonometry calculations are special cases and beyond the intent of this article. They would, however, be of greater interest to the specialized practitioner.

Is That (calculator) Answer Correct? – Notes

(1) *For a discussion of other non-calculator methods that were used before calculators see Calculating Before Calculators in* ***HP Solve*** *Issue 21 page 27.*
http://h20331.www2.hp.com/hpsub/downloads/Newsletters_HP_Calculator_eNL_12_Dec_2010_v1.pdf

(2) *The topic of learning math will be a topic of debate "forever." In a modern complex high tech world we all have to be able to differentiate between a wide variety of measurements. In the engineering world there is a saying that if you can't measure it you can't control it. Measuring requires numbers and numbers requires mathematics.*

(3) *The name "The rule of nines" is used for many purposes including medicine (assessing body burn area), and accounting fraud as well as being the name of a band. In nursing the rule of nines refers to body surface area.*

Calculators and Prefix Units

HP Solve #28 page 40

← Previous Article Next →

Return to top

Calculators and Prefix Units

Richard J. Nelson

Electronics is probably one of the most diverse fields where the range of units extends over a dozen decades of values. I was reminded of this when I had to unpack my electronics components and sort them into bin values. Table 1 illustrates the value range of a very common component, resistors. This range of values is common place, and actually extends several decades in both directions beyond the table in more specialized applications

Table 1 – Common Range of Electronics[1] Resistance Values

Decade	Example	Prefix	EEX	Comments
< 0.0000001	≈ Super conductivity		$10^{-\infty}$	
0.000001 - 0.0001	Materials studies	Micro ohms, μΩ	10^{-6} -10^{-4}	
0.001	Switch contacts	Milli ohms, mΩ	10^{-3}	
0.01	Auto starter wiring			
0.1	House wiring, AA cell			
1		Ohms, Ω	10^{0}	Low end of most Digital Multimeters, DMMs.
10	100 Watt bulb, cold			
100	100 Watt bulb, hot			
1,000		Kilo ohms, kΩ	10^{3}	
10,000	Human body			
100,000				
1,000,000	ESD dissipative	Mega ohms, Ω	10^{6}	
10.000.000				High end of most Digital Multimeters, DMMs.
100,000,000	Hi Voltage dividers			
1,000,000,000		Giga ohms, GΩ	10^{9}	
>1,000,000,000,000	Insulator resistance	Tera ohms, TΩ	10^{12}	

Electronics is like any technical discipline in that certain units tend to be favored in practice. In the mechanical field, for example, a length measurement of millimeters is often used. Bolt or wrench sizes are in millimeters e.g. 18 mm vs. 1.8 cm. In the very early days of calculators that featured English-Metric conversions it was inches-centimeters. Today it is usually inches –millimeters to better match the units in most common use.

Electronic values are similar. 0.01 Amperes is immediately “converted” to 10 mA by those who use these values. The usage of milliamperes even extends to thousands. Battery cell capacity may be rated at 1200 mAh rather than 1.2Ah.

These examples illustrate that in practice engineering units are not always in multiples of 3 as indicated by the unit prefixes, Kilo or milli.

I like the mm vs. cm example to illustrate that the purpose of the calculator is to make the task easy and simple. I know that 1.8 cm is 18 mm. This is a “no brainer,” but when I have my mind wrapped around a problem I find it easier to not have to mentally move that decimal point. I know that 1.2Ah is 1200mAh, but I don’t want to make that mental conversion if I don’t have to because my focus is on other things.

The need for the conversion is that all the other similar values from data sheets, etc. are expressed in mAh rather than Ah. They are equivalent and are only expressed using different prefix values.

Many programs (program libraries) have been written to make electronics related calculations. Resistance values are more common above the basic unit of 1 ohm. Capacitor values are quite different because the unit value is the Farad and 99^{+}% of capacitor values are a very small fraction of a Farad. Capacitor values are most commonly in microfarads, nanofarads, or picofarads. Current values, Amperes, probably have the greatest range, and current may range from femtoamperes to kiloamperes - 18 decades.

The RPL graphing calculators have a great unit management system. It would be even better, however, if the common units could be presented as an input/output menu with the prefix values that are used in practice.

If I were making an ohms law calculation I would like to be able to input voltage in millivolts (or volts or kilovolts, etc.), resistance in megohms (or ohms, kilohms, etc.), and get my answer in microamperes (or milliamperes, etc.). This would make problem solving even more convenient.

An example where this would be especially useful is a problem wherein the prefix units are provided and I am not familiar with them. I could just input the values as I find them.

How could this concept be implemented? Perhaps there could be a function called Prefix which would provide the options that would match the units in use. The program would convert the sub or multiple units to the basic unit, do the calculation, and then convert the basic unit to the unit desired using a Prefix units menu. A 'smart" program would provide a labeled default prefix with a menu option to show it in other prefix units.

This prefix I/O would be applicable to any type of problem and could be a part of the calculator software.

What do you think of this idea?

What kind of technical problems do you solve? Do you use prefix units?

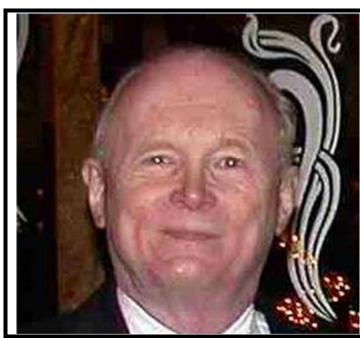
How do you imagine this could be implemented in practice?

Contact me at the email address below.

Calculators and Prefix Units - Note

(1) *Twenty one decades of (prefix) values is represented by the frequencies of the Electromagnetic Radio Frequency Spectrum. The internationally allocated RF frequency is between 9 KHZ (10^3) and 300 GHz (10^9) and the known frequency range is 0 to 3000 EHz (10^{21}). This includes IR, visible light, µv, and X-rays. Tera is 10^{12}, Penta is 10^{15}, Exa is 10^{18}.*

About the Author



Richard J. Nelson, a long time HP Calculator enthusiast, was editor and publisher of *HP-65 Notes*, *The PPC Journal*, *The PPC Calculator Journal*, and the *CHHU Chronicle*. He has also had articles published in *HP65 Key Note* and *HP Key Notes*. As an Electronics Engineer turned technical writer Richard has published hundreds of articles discussing all aspects of HP Calculators. His work may be found on the Internet and the HCC websites at: hhuc.us He proposed and published the PPC ROM and actively contributed to the UK HPCC book, *RCL 20*. His primary calculator interest is the User Interface. Richard may be reached at:

rjnelsoncf@cox.net

From The Editor

← Previous Article Next →

Return to top

From The Editor – Issue 28

Summer is in full force here in the Arizona Sonoran Desert of the Valley of the Sun with triple digit temperatures during the day and high 80's at night. While most people get outside during the summer we do not because the heat here can kill you.

The repurposed HP 30b (aka WP 34S) is an RPN scientific calculator creation the HP user community uses to explore new ideas for the high machine. The HP 20b is also repurposeable but it has been phased out so the 30b is the primary machine to reprogram. Of course the HP warranty is voided. Once the experimental user makes this step into the unknown it is usually a "go for broke" move that is exciting, educational, and often inspirational. Many users have now added a low cost IR diode to their machines and the WP 34S operating system has been modified to be able to use the IR to drive an HP 82240B IR printer.

One aspect of making the conversion is obtaining a serial interface cable to connect the HP calculator to a PC to down load the new operating system. HP made a single batch of these cables and they are nearly exhausted. HHC attendees of the last two Conferences have these cables and I hope that they will share them with others if they are not using them. Eric Rechlin, and others, have these cables and they will offer a repurposed HP 30b to those who want the WP 34S machine but do not want to take the time to set up their own repurposing system.

Here is a list of ***HP Solve*** articles and commentary that will provide additional details.

1. HP 82240B IR Printer Issue 18, May 2010.
2. Repurposing the HP 20b/30b Calculator Platform by Jake Scchwartz Issue 24, July 2011, page 22.
3. From the Editor WP 34S memory up date, etc., Issue 25, October 2011, page 58.
4. HHC 2011 Report Conference coverage, Issue 26, January 2012, pp 10 & 12.
5. The WP 34S Evolves by Jake Schwartz, Issue 27, April 2012, page 34.

If you need a WP 34S Overlay: http://commerce.hpcalc.org/overlay.php This page will also provide the written instructions on how to flash the calculator as well as how to obtain a WP 34S machine.

Here is the content of this issue.

S01 – Graduation Calculators Summer is a time for finishing one phase of academic life and moving to a higher phase. HP calculators for this forward move are featured.

S02 – The China Project There are a lot of students in China that need HP calculators. Here is a report on how HP is working with the China Ministry of Education to introduce graphing calculators and data collection technologies to these students.

S03 – Ostrowski's Method for Finding Roots by Namir Shammas explores an improved Newton's method for finding roots that all students learn in school. The Ostrowski method is examined, explained and exampled with programs for the HP 39gII and the repurposed HP-30b (aka WP 34s). Here are typical examples.

$f(x) = x^3 + 4x^2 – 15$
$f(x) = x \exp(x^2) – \sin^2(x) + 3\cos(x) + 5$
$f(x) = \sin(x) – x/2$
$f(x) = 10 x \exp(-x^2) – 1$
$f(x) = \cos(x) – x$
$f(x) = \sin^2(x) – x^2 + 1$
$f(x) = \exp(-x) + \cos(x)$

S04 – Store and Recall on HP Calculators – the First Decade Calculator memory always seems limited because as improved technology provides additional memory more complex problems become calculator solvable. Here is a historical review of how HP has managed the memory usage of the calculators of the first decade. The decisions made in the 70's still impact the calculators of today.

S05 – Measuring Calculator Current You press the ON key and nothing happens; now what? You replace the batteries and you still have a blank display. Do you toss the machine and buy a new one? The most powerful technique to knowing what is or is not going on in the calculator is to monitor the current it draws by measuring the calculator current. The meter is not expensive, as low as $3, so why not check it out?

S06 – Is that (calculator) Answer Correct? How do you know that the answer you have just calculated is correct? A few easy to learn (or review) techniques may be applied to approximate the answer so that you could even get by without the machine if you had to. You need to know that all those numbers in the display are "correct."

S07 – Calculators and Prefix Units. Calculators must be easy to use. Here is an idea for making them even easier and more convenient for problem solving. What do you think of this idea?

S08 – Regular Columns

- **From the editor.**
- **One Minute Marvels.**
- **HP User Community News; HC 2012, The Calculators of Hewlett Packard Poster, A Bit of HP History – PH or HP?**

S09 – Calculator Usage – Is Efficiency Important? One of the reasons calculators are still so popular considering the many other devices we use every day that could converge the calculator into them is that calculators are so efficient. This article reviews what this means and dispels some of the arguments often made to discount calculator usage.

That is it for this issue. I hope you enjoy it. If not, tell me!

Also tell me what you liked, and what you would like to read about.

X < > Y,

Richard

Email me at: hpsolve@hp.com

HP 48 One Minute Marvels No. 15 – Two alarms

One Minute Marvels, OMMs, are short, efficient, unusual, and fun HP 48 programs that may be entered into your machine in a minute or less. These programs were developed on the HP 48, but they will usually run on the HP 49 and HP 50 as well. Note the HP48 byte count is for the program only.

Two Alarms

Here are two simple repeating alarm sounds programmed to start when you press the menu key and stop when you press any other key. **'ALM1'** and **'ALM2'** produce alternate low, then high, tones. **'ALM2'** is about twice as fast as **'ALM1'**. **'AM1'** repeats about once per second.

‘ALM1’ << DO 900 .2 BEEP .2 WAIT 1200 .2 BEEP .4 WAIT UNTIL KEY END DROP >>

15 commands, 95.5 Bytes, # C3Afh

‘ALM2’ << DO 1000 .1 BEEP .1 WAIT 500 .1 BEEP .2 WAIT UNTIL KEY END DROP >>

15 commands, 95.5 Bytes, # FD5h

Both programs are structured the same using a DO Loop to repeat until a test is made. The number following the DO is the frequency in hertz. The second number is the duration in seconds. These two numbers are the arguments for the BEEP command which produces the sound. The number following the BEEP is the argument for the WAIT command which will wait for the duration in seconds. A second BEEP-Wait sequence follows for the second tone. In the case of **‘ALM1’** a higher tone follows the lower tone which is the reverse order of tones of **‘ALM2’.** The end of the DO clause is the UNTIL command which is followed by the KEY command. The Key command will cause the loop to be terminated when a key is pressed. The END is then executed and the DO Loop is terminated. The last command, DROP, is used to remove the unused key code that the KEY command places on the stack.

HP User Community News

HHC 2012

The final arrangements for HHC 2012 were not quite finalized as this issue was being prepared to be mailed and posted. A venue change was made at the last minute. It is good that the HHC Committee had a “Plan B.” The Conference this year will be held in Nashville TN on September 22 & 23. The hotel and travel details will be posted - hopefully by the time you read this - at: hhuc.us

The Calculators of HP Poster

Rick Furr is an HP calculator fan. Like most fans he has a website – vcalc.net. A few HP fans get so enthusiastic about their hobby that they apply their knowledge and experience towards a calculator related “project.” In the 38 years that I have been writing about HP calculators I have reported on hundreds of such projects. In Rick’s case it is assembling high resolution calculator images into a poster format.

Seven years ago Rick printed a 10 row poster of all of HP’s calculators. He has now updated that poster with a new 12 row one that is about as complete as you can get in having all vital information related to all HP calculators in one place. The new information included in the revision is extensive.

Fig. 1 and Fig. 2 show the first HP calculator and the most recent calculator respectively. Fig. 3 shows a low resolution overview of the poster to give you some idea of what the 18” x 24” size contains.

The approach most enthusiasts use is to put as much information as possible in the space available. Since the aspect ratios of HP machines varies, it is not possible to arrange them in neat regular rows and columns. The odd space “must” be used and many calculator related items are also included. I made a poster image count and here is what I discovered – in addition to the poster source, two HP desktop tables and a legend.

Table 1 – Poster Image Categories

Calculators (unique models) . .	107
Palmtops	4
Accessories	8
Misc.	8
Non-calculator (SM 400, 970)	2

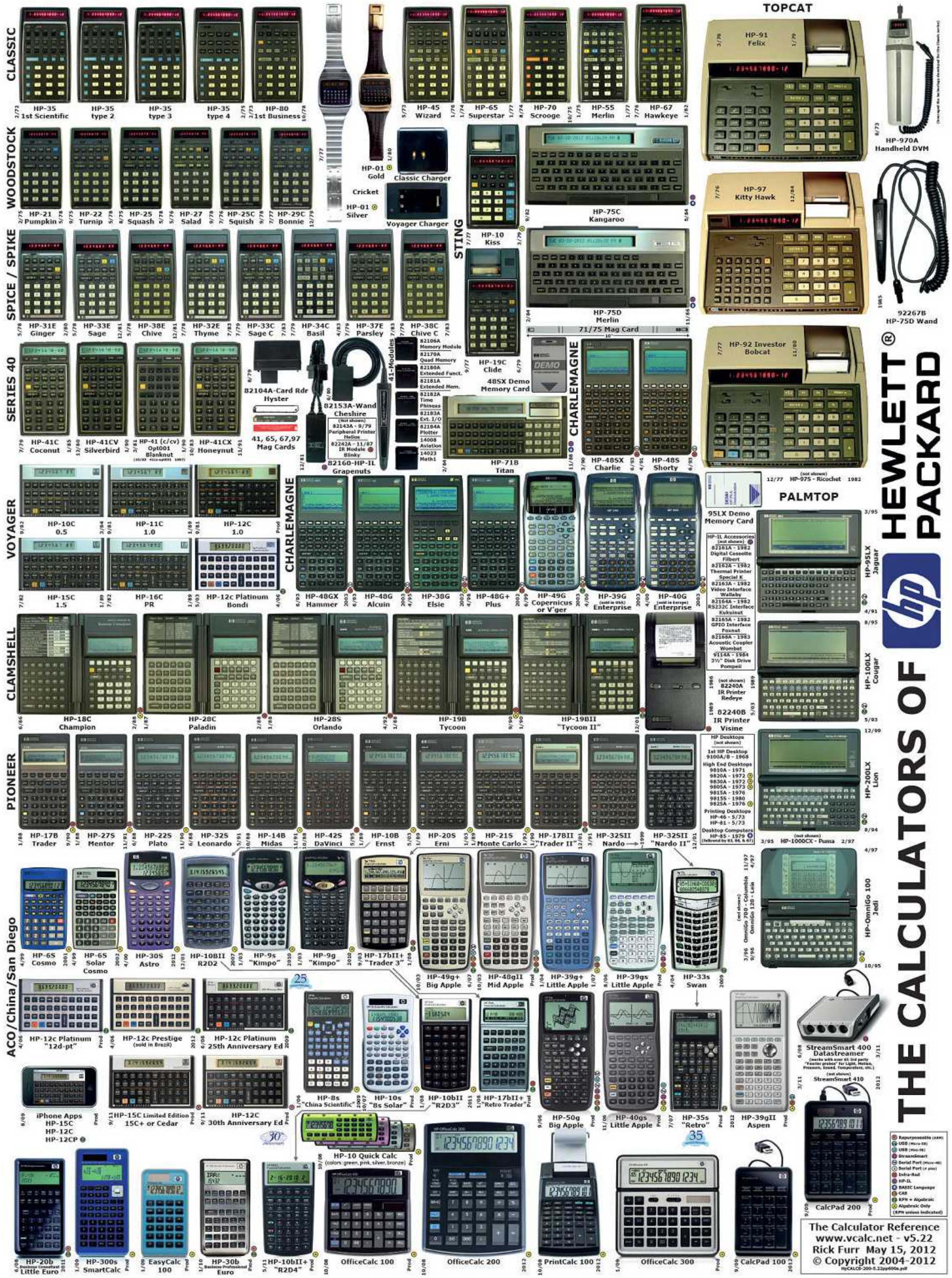
Fig. 1 – Four variations of the 1st HP calculator are shown on the poster on the upper left. (Type 1 shown).

Fig. 2 – The most recent HP calculator is shown at the bottom right. Prod means it is current production.

The poster calculators are of two orientation types, vertical format (85), and horizontal format (22). Horizontal format calculators are 20.6% of all HP calculators made. Because the Palmtops were an outgrowth of HP's calculators it, is natural to include them. An example of the accessories and miscellaneous images is shown in Fig. 4. The I/O and OS legend is shown in Fig. 5. Fig. 2 illustrates the use of the Legend in the right bottom corner.

The information for each calculator model provided on the improved poster includes the following.

High resolution photograph – keyboards are readable with a magnifying glass.
Code name of the calculator – some are numbers.
Introduction date of the calculator.
Discontinuance date of the calculator.
The family name or grouping classification of each calculator.
The operating system/logic system of each calculator.
The I/O options of each calculator.
Version information of special calculators, e.g. 25, 30, or 35th anniversary versions.
Display representation in every image of every calculator.
Individual images of variations of special machines, e.g. four types of the HP-35A.
Calculators of same model number are noted with connecting arrows.



CLASSIC
HP-35 1st Scientific
HP-35 type 2
HP-35 type 3
HP-35 type 4
HP-80 1st Business
HP-45 Wizard
HP-65 Superstar
HP-70 Scrooge
HP-55 Merlin
HP-67 Hawkeye
TOPCAT
HP-91 Felix
HP-970A Handheld DVM
WOODSTOCK
HP-21 Pumpkin
HP-22 Turnip
HP-25 Squash
HP-27 Salad
HP-25C Squish
HP-29C Bonnie
HP-01 Gold
HP-01 Silver
Classic Charger
Cricket
Voyager Charger
HP-75C Kangaroo
HP-97 Kitty Hawk
STING
HP-10 Kiss
SPICE / SPIKE
HP-31E Ginger
HP-33E Sage
HP-38E Chive
HP-32E Thyme
HP-33C Sage C
HP-34C Basil
HP-37E Parsley
HP-38C Chive C
HP-75D Merlin
71/75 Mag Card
92267B HP-75D Wand
SERIES 40
82104A-Card Rdr Hyster
82153A-Wand Cheshire
41-Modules
HP-19C Clide
48SX Demo Memory Card
HP-92 Investor Bobcat
HP-41C Coconut
HP-41CV Silverbird
HP-41 (c/cv) Opt001 Blanknut
HP-41CX Honeynut
41, 65, 67,97 Mag Cards
82160-HP-IL Grapenuts
HP-71B Titan
CHARLEMAGNE
HP-48SX Charlie
HP-48S Shorty
HEWLETT PACKARD
VOYAGER
HP-10C 0.5
HP-11C 1.0
HP-12C 1.0
HP-15C 1.5
HP-16C PR
HP-12c Platinum Bondi
HP-48GX Hammer
HP-48G Alcuin
HP-38G Elsie
HP-48G+ Plus
HP-49G Copernicus or V'ger
HP-39G Enterprise
HP-40G Enterprise
PALMTOP
95LX Demo Memory Card
HP-95LX Jaguar
HP-100LX Cougar
CLAMSHELL
HP-18C Champion
HP-28C Paladin
HP-28S Orlando
HP-19B Tycoon
HP-19BII "Tycoon II"
82240B IR Printer Visine
HP-200LX Lion
PIONEER
HP-17B Trader
HP-27S Mentor
HP-22S Plato
HP-32S Leonardo
HP-14B Midas
HP-42S DaVinci
HP-10B Ernst
HP-20S Erni
HP-21S Monte Carlo
HP-17BII "Trader II"
HP-32SII Nardo
HP-32SII "Nardo II"
HP-1000CX - Puma
HP-OmniGo 100 Jedi
THE CALCULATORS OF
San Diego
HP-6S Cosmo
HP-6S Solar Cosmo
HP-30S Astro
HP-10BII R2D2
HP-9s "Kimpo"
HP-9g "Kimpo"
HP-17bII+ "Trader 3"
HP-49g+ Big Apple
HP-48gII Mid Apple
HP-39g+ Little Apple
HP-39gs Little Apple
HP-33s Swan
ACO/China
HP-12c Platinum "12d-pt"
HP-12c Prestige (sold in Brazil)
HP-12c Platinum 25th Anniversary Ed
HP-8s "China Scientific"
HP-10s "8s Solar"
HP-10bII "R2D3"
HP-17bII+ "Retro Trader"
HP-50g Big Apple
HP-40gs Little Apple
HP-35s "Retro"
HP-39gII Aspen
StreamSmart 400 Datastreamer
iPhone Apps HP-15C HP-12C HP-12CP
HP-15C Limited Edition 15C+ or Cedar
HP-12C 30th Anniversary Ed
HP-10 Quick Calc (colors: green, pink, silver, bronze)
HP-20b Little Euro
HP-300s SmartCalc
EasyCalc 100
HP-30b Euro
HP-10bII+ "R2D4"
OfficeCalc 100
OfficeCalc 200
PrintCalc 100
OfficeCalc 300
CalcPad 100
CalcPad 200
The Calculator Reference
www.vcalc.net - v5.22
Rick Furr May 15, 2012
© Copyright 2004-2012

Fig. 4 – Example of Table 1 Accessories and Miscellaneous categories.

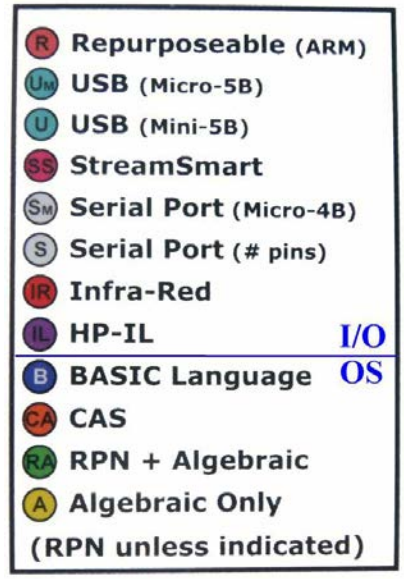


Fig. 5 – I/O and OS Legend.

One calculator is not shown – but is mentioned – the HP-97S with its special I/O. A special frequency modulation printing process makes this poster of exceptional quality. While this description is quite complete, there are still discoveries to be made by studying the poster.

The poster cost is $19.95 plus $5.05 for shipping. One poster with shipping is $25.00 Shipping is by USPS Priority in the US and Airmail postage to Canada or Mexico. See the website for shipping to other countries. The poster makes a unique gift for any HP Calculator collector.

http://www.vcalc.net/hp.htm

Fig. 5 – The only images not having readable legends.

Rick also has a few other unique posters. See his website for the details. His order processing is first class and his shipping is speedy on receipt of order. Special pricing and reduced shipping is obtained for multiple poster orders.

A Bit of HP History - PH or HP?

Every now and then I run across an article on the Internet that reminds me of the past. With HP having grown so large and the founders that gave the company its name long gone I thought that I would reprint a few words I wrote over 37 years ago. In memory of Bill and Dave.

From "65 Notes", V2 N1 P3 (January 1975)

In January 1939, two Stanford Electrical Engineering graduates flipped a coin to see if their partnership would be called Packard-Hewlett or Hewlett-Packard. David Packard lost on that flip, and Bill Hewlett had his name first. The occasion was the receipt of the first big order from Walt Disney Studios for nine audio oscillators called the Model 200A. The early experimental workshop was literally a 'garage' operation. In 1947 the firm incorporated and annual sales reached $1.5 million. New products were added to the expanding product line of signal generators, microwave instruments, amplifiers, etc., and by the '50's the company was developing instruments at the rate of 20 new ones per year.

The strong leadership exerted by Bill Hewlett and David Packard is what has made Hewlett-Packard the unique company it is today. Another key factor has been people - the employees. Over 20,000 people work for Bill and Dave, who hold 51% of the company stock. The company operates on policies that strongly reflect the philosophy of the two men. The open and free exchange of information among engineers and designers is encouraged by not allowing offices and cubicles. Even today, employees feel that they are part of a family. When times are good, the family shares the prosperity through profit sharing checks, liberal fringe benefits, etc. When the economy is down, Hewlett-Packard does not lay off its employees (who know the H-P philosophy, and will work a 4-day week if asked to).

In March 1968 Hewlett-Packard introduced the HP-9100A table-top programmable calculator. At that time Bill Hewlett wondered if the next calculator developed could not be a tenth the size and cost of the 9100A. Later the goal was formalized to be a series of ten machines to be hand-held, battery operated, and capable of being carried in his shirt pocket - which was measured on the spot. Thus, the HP-35, -80, -45, -81, -46, -65, -70 and -55 were born, along with machines yet to be announced. This brainstorming effort was being carried out in earnest by the newly-formed Advanced Products Division by the fall of 1970. The technological accomplishment was in getting the HP-35 to market, from concept to announcement, in less than 18 months.

Operating by the Philosophy of providing technically advanced products of high quality, based on excellent executive direction, Hewlett-Packard continues to be the leader in the electronics industry. Traditionally, H-P has not sold its products to the general public.

The Advanced Products Division, with its pocket calculator, was the first market of this type that the company entered. When H-P did enter the calculator market, it did not compete with existing manufacturers; it created a completely new market - the scientific calculator market.

References:

1. The HP-35: A Tale of Teamwork with Vendors, by Gerald M. Walker, Electronics, February 1, 1973.
2. Hewlett-Packard Company Annual Report, 1973.
3. A New Electronic Calculator with Computer-Like Capabilities, by Richard E. Monnier.

About the Editor

Richard J. Nelson, a long time HP Calculator enthusiast, was editor and publisher of *HP-65 Notes*, *The PPC Journal*, *The PPC Calculator Journal*, and the *CHHU Chronicle*. He has also had articles published in *HP65 Key Note* and *HP Key Notes*. As an Electronics Engineer turned technical writer Richard has published hundreds of articles discussing all aspects of HP Calculators. His work may be found on the Internet and the HCC websites at: hhuc.us He proposed and published the PPC ROM and actively contributed to the UK HPCC book, *RCL 20*. His primary calculator interest is the User Interface. Richard may be reached at: rjnelsoncf@cox.net

Calculator Usage — Is Efficiency Important?

HP Solve #28 page 51

← Previous Article

Return to top

Calculator Usage – Is Efficiency Important?

Richard J. Nelson

Introduction

In recent years the usage of, or need for, a calculator has been questioned. Students are required to have a computational tool, and business users need special business applications "in the field." Cell phones and smartphones have calculator applications "built in," and most of the other personal electronic devices such as handhelds, tablets, and even laptops include the ability to load just about any calculator model of choice as an application. Integrating calculator applications into "other" more popular electronics devices is frequently talked about using the term convergence. Convergence is often used as an example of the death knell for calculators. Yet year after year, calculators continue to be sold by the tens of millions.

I personally believe that one of the major reasons calculators continue to be popular is that they are inexpensive, easy to use, and efficient. They instantly turn on, they have well designed keys and menus - and they are very efficient.

Efficiency

Efficiency is an objective term. It is not an opinion or a judgment. It is defined as that ratio of input to output expressed as a percentage, the higher (100% is highest) the better. While the application of this ratio is not usually applied in the testing or evaluation of calculators, a comparative number of two machines, or systems, are frequently used. The most common "efficiency" comparison is keystroke counting. The argument is that solving a problem with three keystrokes is more efficient than solving it with five keystrokes[(2)]. How important is this? Obviously it is, but there are "exceptions."

One argument is made that keystroking time is irrelevant in that even for the highest-paid-calculator user the keystroke saving time is so small as to be deep in the user time usage noise. Time wasted on other non-work related "human" activity – telling jokes, drinking coffee, etc. – is far more wasteful and inefficient. This perspective poses that even thinking about the time saved by more efficient keystroke sequences is just plain silly.

Another perspective related to efficient calculator usage is speed, or the time spent on solving a problem.

It's all about being human

Calculator speed is like driving a car. Some (most?) prefer to drive fast. Why is this? I am sure that there are many studies to explain the obvious that include such factors as male/female, emotional state, being late, owning a fast car, etc. I will offer what I consider is an important consideration – boredom.

When human beings are seriously working on a problem, they tend to focus on "getting it done." The computational part of the problem is done by the machine, so in the mind of the problem solver calculator speed is a process step that is out of the calculator user's control. If the problem involves many calculations or repetitive calculations, the calculator user gets a bit frustrated if everything doesn't go smoothly and quickly.

I was using a non-HP calculator "clone" because that was all that was available at the time and place I was "problem solving." I was exploring a number of electronics circuit configurations[(1)] and the calculator had a terrible keyboard. I was forced to press each key and visually check the display to see if the key responded. Since I wanted the full ten digit accuracy I was pressing a large number of keys for just a few dozen calculations. The calculator was clearly getting in the way of problem solving. Is keystroke counting important? Yes it is. Is the calculator response time important? Yes it is.

A program is always faster

The machine I had to use was an RPN keystroke programmable and I entered a program to save time while making notations on a napkin. In terms of HP-15C Programming (typical RPN operations and structure) I thought of how the program should be approached. The program starts assuming that the two variables (A & B are used for R1 & R2[1]) are on the stack as shown in the START column.

My first program sketched on a napkin was the most general straight forward method of storing the variables in a register and recalling them as needed. In this example each variable is needed twice. See Fig. 1 below. This is the easiest to program and stack usage is simplest with only a concern that the stack is not "overloaded." This approach required nine program steps.

Step	1	2	3	4	5	6	7	8	9	10
T	~	~	~	~	~	~	~	~	~	~
Z	~	~	~	~	~	~	A×B	~	~	~
Y	A	A	B	B	~	A×B	A	A×B	~	~
X	B	B	A	A	A×B	A	B	A + B	A×B/A + B	A×B/A + B
press	START	STO2	X⇄Y	STO1	×	RCL1	RCL2	+	÷	R/S

Fig. 1 – Most straight forward programming method stores variables and recalls them as needed – 9 steps.

My second program utilized the LASTX register in order to save a register because most calculators work faster if the calculations are done on the stack. See Fig. 2 below. This saved one program step.

Step	1	2	3	4	5	6	7	8	9
T	~	~	~	~	~	~	~	~	~
Z	~	~	~	~	~	A×B	~	~	~
Y	A	A	B	~	A×B	A	A×B	~	~
X	B	B	A	A×B	A	B	A + B	A×B/A + B	A×B/A + B
press	START	STO1	X⇄Y	×	LASTX	RCL1	+	÷	R/S

Fig. 2 – "Improved" program using LASTX to avoid using a storage register – 8 steps saves one step.

The third and final program uses the stack for all calculations by using the R↑ operator and auto T register replication. LASTX is considered part of the stack. See Fig. 3 below. This program uses nine program steps.

Step	1	2	3	4	5	6	7	8	9	10
T	~	~	A	B	B	B	B	B	B	B
Z	~	A	B	B	B	B	A×B	B	B	B
Y	A	B	B	B	B	A×B	A	A×B	B	B
X	B	B	B	A	A×B	A	B	A + B	A×B/A + B	A×B/A + B
press	START	↑	↑	R↑	×	LASTX	R↑	+	÷	R/S

Fig. 3 – Program variation to make all calculations on the stack for maximum speed – 9 program steps.

Note that all programs end with Run/Stop, R/S, so that the R/S key is all that is needed to run the program. This works because it was the only program in memory. I concluded that the Fig. 3 program is the most efficient way to use the machine for making these calculations under these conditions.

Efficiency vs. skill

Keystroke counting[3] is logically an efficiency measure for comparing calculators. The use of the calculator, however, is by a human being and there is a human quality that may actually negate the "fewer keystroke" argument under special conditions. If you repeat a sequence of key presses many times, you may learn this sequence by rote. If the sequence is five keystrokes and you are able to press these five keys faster than pressing a more efficient three keystroke sequence, your skilled method may be more effective. Dr. Wickes, HP's RPL Team leader used to ask this question. How do you differentiate between familiar and friendly? For the average user, however, using fewer keystrokes is better and more efficient.

Other efficiency issues

Any comparison between two calculators that makes the machine easier to use could be considered an efficiency issue because the result is a savings of time. Issues such as keyboard colors used, display readability, learning curve[3], reliability, keyboard layout, and even accuracy all contribute to the calculator efficiency issue.

Conclusion

The calculator continues to be important to users in spite of convergence because dedicated calculators are inexpensive, easy to use, and efficient. Efficiency is defined and how calculator efficiency may be compared with the most common contrary arguments explained. Since efficiency is an objective measurement, the conclusion that the answer to the title question has to be yes. Efficiency is a vital part of the calculator's advantage in today's technical environment.

Calculator Usage – Is Efficiency Important? – Notes

(1) *The calculation was for two parallel resistors which uses "the product divided by the sum"* $\frac{R1 \times R2}{R1+R2}$ *The problem was evaluating an infinite ladder resistor network stage by stage.*

(2) *Keystroke counting is more effective if done to the nearest 1/2 keystroke. Since each keystroke requires a search time and a press time, there are situations where the same key is pressed twice. This would only be 1-1/2 keystrokes.*

(3) *You may further explore the issue of HP calculators in more detail by reading an article published for HHC 2006. The article is a dialog among six experienced users discussing the topic of* *__Learning to Use HP Calculators__. See: http://hhuc.us/2006/LearningToUseHPCalcs.pdf*